High Multiplicity Clustering Using the Barrel Electromagnetic Calorimeter at STAR

Darrick Jones – The College of New Jersey
Cyclotron Institute REU 2009
Advisor: Dr. Saskia Mioduszewski
Outline

- Quark Gluon Plasma (QGP)
- Jets
- STAR – BEMC
- Clustering Algorithm
- Modularization
Quark Gluon Plasma

- Matter so hot (>10^{12} K) and dense that quarks and gluons (partons) are no longer confined to hadrons.
- Recreated in heavy ion collisions at RHIC.
- Believed to have existed up to 10^{-5} s after the Big Bang.
- Too short lived to be observed directly.
- Must be investigated indirectly.

Ordinary nuclear matter
protons and neutrons

Quark-Gluon Plasma
quarks and gluons

Heat and compress
Hadron-Triggered Jets

- Large transfers of energy cause partons to scatter.
- Hard-scattered partons fragment into cones of hadrons forming back-to-back “jets”.
- Scattering occurs before matter is created.
- Hard-scattered partons traverse through and probe matter created.
- As they pass through the medium, they interact and lose energy.
Jet produced back-to-back with “direct” photon.
Photon maintains initial energy as it passes through the medium.
Initial energy of parton and photon can be considered equal.
By measuring energy loss of the jet, it can be determined how the parton was affected by the medium.
\( \pi^0 \) Decay

- Direct photons must be distinguished from background photons.
- Main source of background is \( \pi^0 \) decay.
- Primary decay mechanism:
  \[ \pi^0 \rightarrow \gamma + \gamma \]
- Heavy ion collisions create an environment with lots of background.
- Need a clustering algorithm designed for high multiplicity.
STAR detector offers a unique myriad of subsystems which can be utilized to distinguish decay photons from direct photons.
Geometry of STAR

Beam View

Side View

Pseudo-rapidity: $\eta = -\ln[\tan(\theta/2)]$
**Barrel Electromagnetic Calorimeter (BEMC)**

- Designed to record energy deposited from electromagnetic particles.
- 120 modules – 60 in $\phi$ by 2 in $\eta$.
- Each module is divided into 40 towers.
- Towers are 20 layers of lead and 21 layers of scintillator.

- Barrel Shower Max Detector (BSMD) is located after 5 layers of lead and scintillator.
- BSMD further divides each module into 150 strips in $\eta$ and $\phi$, 18000 subdivisions in each direction.
- High spatial resolution allows precise determination of particle position necessary for high multiplicity clustering.
How the Algorithm Works

- Scans $\phi$ and $\eta$ strips for peak energy above threshold.
- Identifies the tower which contains these strips.
- Checks total tower energy.
- Determines exact location of the hit using the BSMD.
- Finds 8 neighboring towers.
- Compares position to find nearest neighbor.
- Based on location of hit, uses strip data from 1 or 2 towers.

For clusters with 8 GeV or more, 65%...
Shower Profile

- From the information gathered by the strips, a transverse shower profile is generated with high enough resolution that decay photons can be separated from direct photons (from simulated particles).

- Useful for rejecting background in the $\gamma$-jet Analysis
Code Modularization

- While the algorithm works very well, the program itself required substantial work.

- Program was not suitable for general use.

- The program needed to be made flexible, readable, and in compliance with STAR coding standards through modularization.
Improvements

- Able to make significant improvements using simple computer programming ideas.
  - Header files hide variables.
  - Structures of variables are designed to allocate variables.

```c
#include "StBmcUtilInfo.h"

struct BmcUtil //class designed to contain BEMC utility
{
    float energy, theta, pedestal, rms, angle, x, y, z,
    qParentTrackP, qParticleId, id, tower, sub,
    status; unsigned int adc, module, eta;
    bool central, nextHiEtq(0), nextHiEq(0),
    nextPedq(0), nextRMsq(0), nextSpq(0),
    pedpq1(0), pednq1(0), pedpq19(0),
    RMSpq1(0), RMSnq1(0), RMSpq19(0),
    Int_t spql(0), snql(0), spq19(0), snq19(0),
    int towerpql(0), towerpq1(0), towerpql(0),
    int Idpq1(0), Idnq1(0), Idpq20(0), Idnq20(0),
    unsigned int adcpql(0), adcnq1(0),
    float thetapq1(0), thetanq1(0), thetapq21(0),
    float ppq1(0), pnq1(0), ppq20(0), pnq20(0),
    float pidpq1(0), pidnq1(0), pidpq20(0),
    UInt_t Modpq1(0), Modnq1(0), Modpq20(0),
    Int_t subpq1(0), subnq1(0), subpq20(0),
    UInt_t etapq1(0), etanq1(0), etapq20(0),
    int nuOfNiq(0), nextIdq(0), nextIdq(0),
    nextHiEtq(0), nextHiEq(0),
    float nextPedq(0), nextRMsq(0), nextSpq(0),
    float pedpq1(0), pednq1(0), pedpq19(0),
    float RMSpq1(0), RMSnq1(0), RMSpq19(0),
    Int_t spq1(0), snq1(0), spq19(0), snq19(0),
};
```
Improvements

• Increase readability and functionality by using predefined functions available in STAR libraries.

Before

```java
if(abs(didT-diQ) == 0 || abs(didT-diQ) == 180 || abs(didT-diQ) == 210 || abs(didT-diQ) == 230 || abs(didT-diQ) == 250) continue;
if(abs(didT-diQ) == 0 || abs(didT-diQ) == 180 || abs(didT-diQ) == 210 || abs(didT-diQ) == 230 || abs(didT-diQ) == 250) continue;
if(eThq < 0.05 && fabs(eT - fabs(eThq)) = 0) continue; if(eThq < 0.05 && fabs(eT - fabs(eThq)) = 0) continue;
if(fabs(eT - fabs(eThq)) > 0.09 || fabs(eT - fabs(eThq)) = 0.09) continue;
if(fabs(eT - fabs(eThq)) = 0.05 && fabs(eT - fabs(eThq)) = 0.05) continue;
if(fabs(eT - fabs(eThq)) = 0.05 && fabs(eT - fabs(eThq)) = 0.05) continue;
if(fabs(eT - fabs(eThq)) = 0.05 && fabs(eT - fabs(eThq)) = 0.05) continue; // add here to these two conditions tower
if(fabs(eT - fabs(eThq)) = 0.05 && fabs(eT - fabs(eThq)) = 0.05) continue;
if(fabs(eT - fabs(eThq)) = 0.05 && fabs(eT - fabs(eThq)) = 0.05) continue;
if(fabs(eT - fabs(eThq)) = 0.05 && fabs(eT - fabs(eThq)) = 0.05) continue;
```

After

```java
StEmcPosition* mEmcPosition = new StEmcPosition();

for(a = -1; a < 2; a++)
{
    for(b = -1; b < 2; b++)
    {
        if(a = 0 && b = 0) continue;
        tower[a+1][b+1].id = mEmcPosition->getNextTowerId(tower[1][1].id, b, a);
    }
}
```

delete mEmcPosition;
Improvements

- Reduced redundancy by using loops and arrays.
Improvements

- By incorporating easily understandable variable names, as well as a structure, and adding member functions:
  - Gave the user the option to define threshold values by eliminating hard-coded values
  - Eliminated redundant loops.
  - Made code more readable.
  - Eliminated or replaced 200+ variables.
  - Decreased significantly the total number of statements.
Further Improvements

- Increase modularization by introducing functions which can be easily moved and replaced if needed.
- Add more flexibility by providing functions to set analysis-specific parameters.
- More of the same.
- Ensure further compliance with STAR coding standards for eventual implementation into the STAR library.
Acknowledgements

- Saskia Mioduszewski
- Ahmed Hamed
- Sherry Yennello
- Larry May
- REU Group
- Texas A&M Cyclotron Institute
- NSF