

# Cross Validation

Pourya Vakilipourtakalou

August 5 - 2016

# Overview

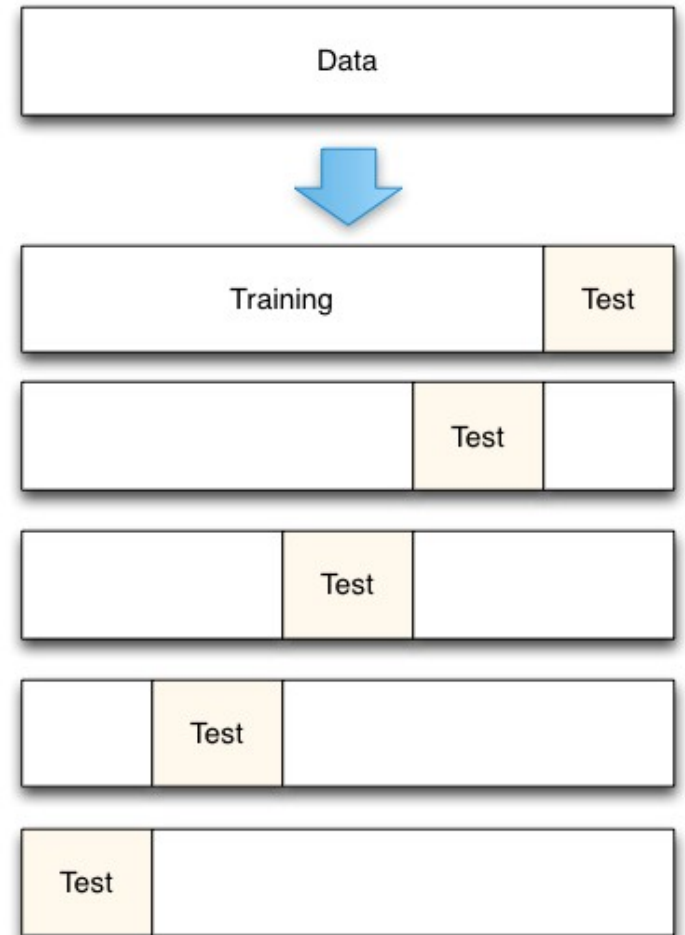
Crossvalidation :

ROC Plots

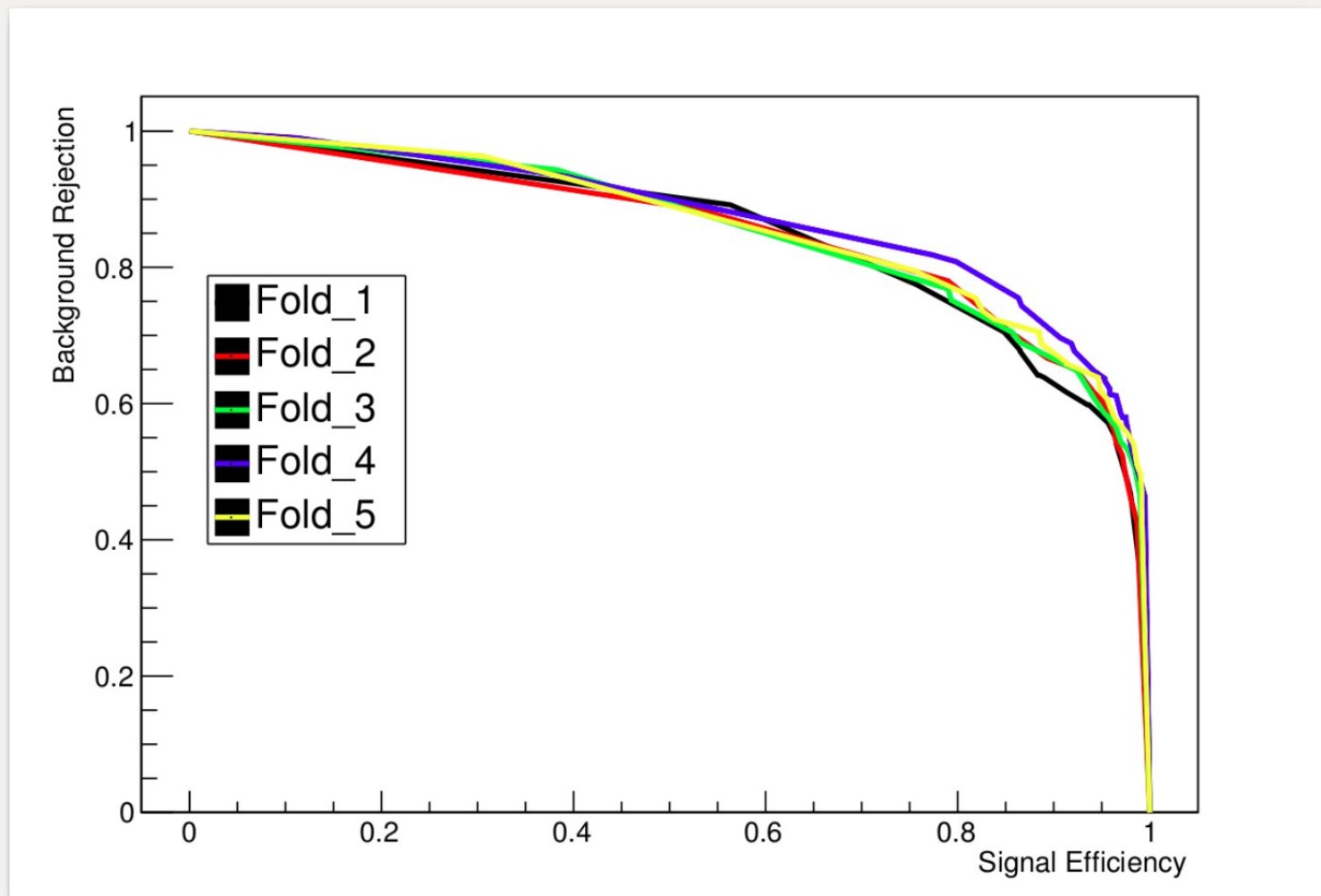
Parallelization

# Cross validation

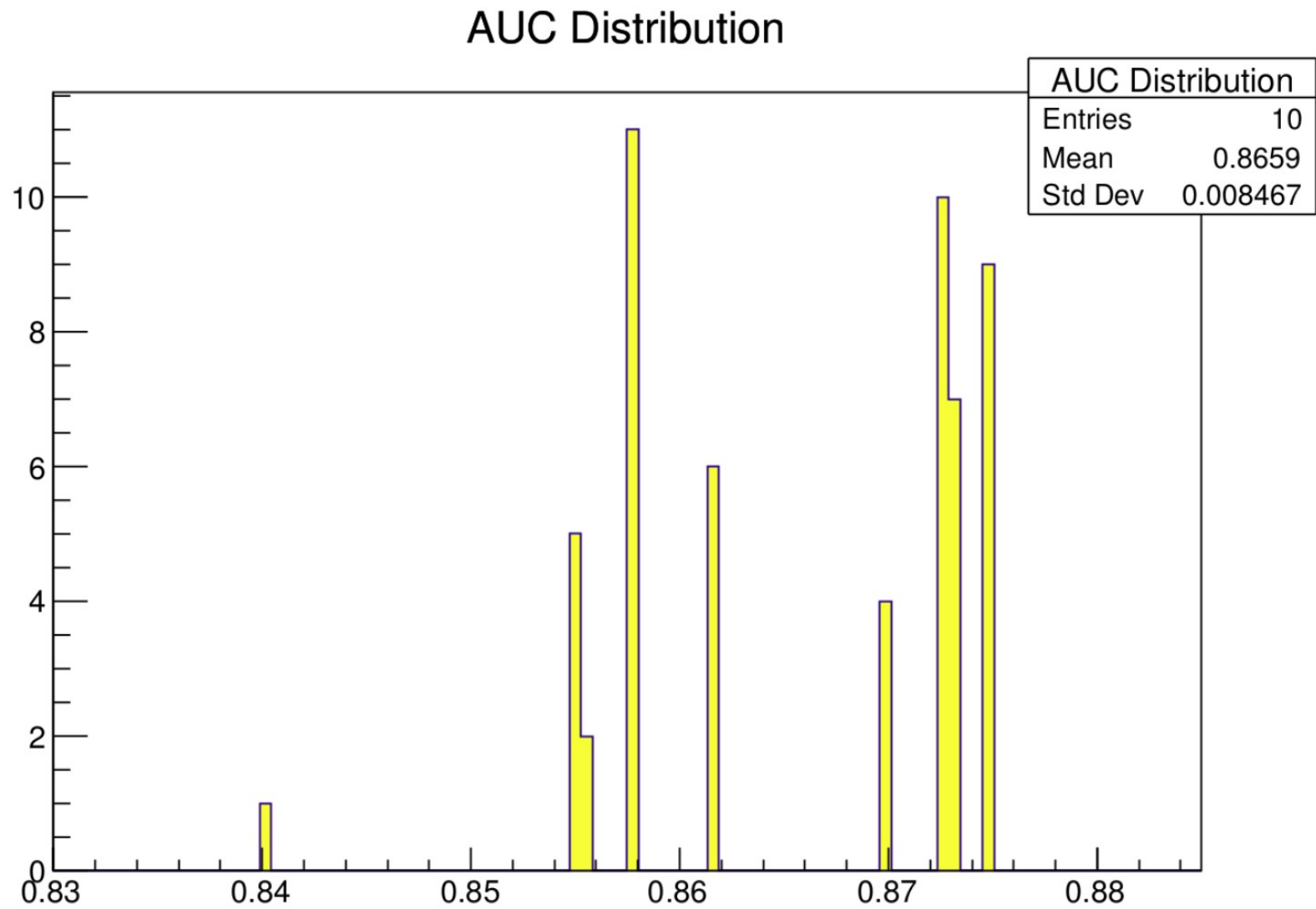
- For each part we have a ROC curve and as a result and AUC integral.



# First Method : PlotROC



# Second Method : AUCDistribution



# Parallelization

- ThreadPool → Multithreading
- TProcPool → Multiprocessing
- Multithreading needs to lock the threads
- Multiprocessing is in some cases slower

```

auto ROCs = pool.Map([=](Int_t j) -> double {

    TString foldTitle = methodTitle;
    foldTitle += "_fold";
    foldTitle += j+1;

    loader->PrepareTrainingAndTestTree(j, TMVA::Types::kTesting);

    TMVA::DataLoader * seedloader = new TMVA::DataLoader(foldTitle);

    for(int index = 0; index<nbits; index++){
        seedloader->AddVariable(varNames.at(index), 'F');
    }

    VIDataLoaderCopy(seedloader, loader);

    BookMethod(seedloader, theMethod, methodTitle, theOption);

    TrainSingleMethod(foldTitle);
    TestSingleMethod(foldTitle);
    EvaluateSingleMethod(foldTitle);

    Double_t rocint ;
    rocint = GetROCIntegral(seedloader->GetName(), methodTitle);

    TMVA::MethodBase * smethod = dynamic_cast<TMVA::MethodBase*>(fMethodsMap[seedloader->GetName()][0][0]);
    TMVA::ResultsClassification * sresults = (TMVA::ResultsClassification*)smethod->Data()->GetResults(smethod->GetMethodName(), Types::kTesting, Types::kClassification);
    sresults->Clear();
    sresults->Delete();
    delete sresults;
    fgTargetFile->cd();
    fgTargetFile->Delete(seedloader->GetName());
    fgTargetFile->Delete(Form("%s;1", seedloader->GetName()));
    fgTargetFile->Flush();
    gSystem->Exec(Form("rm -rf %s", seedloader->GetName()));

    this->DeleteSingleMethod(foldTitle);

    fMethodsMap.clear();

    return rocint;}, ROOT::TSeqI(NumFolds));

//s.Print();
//cout<<endl<<endl;
float sumFOM = 0.0;

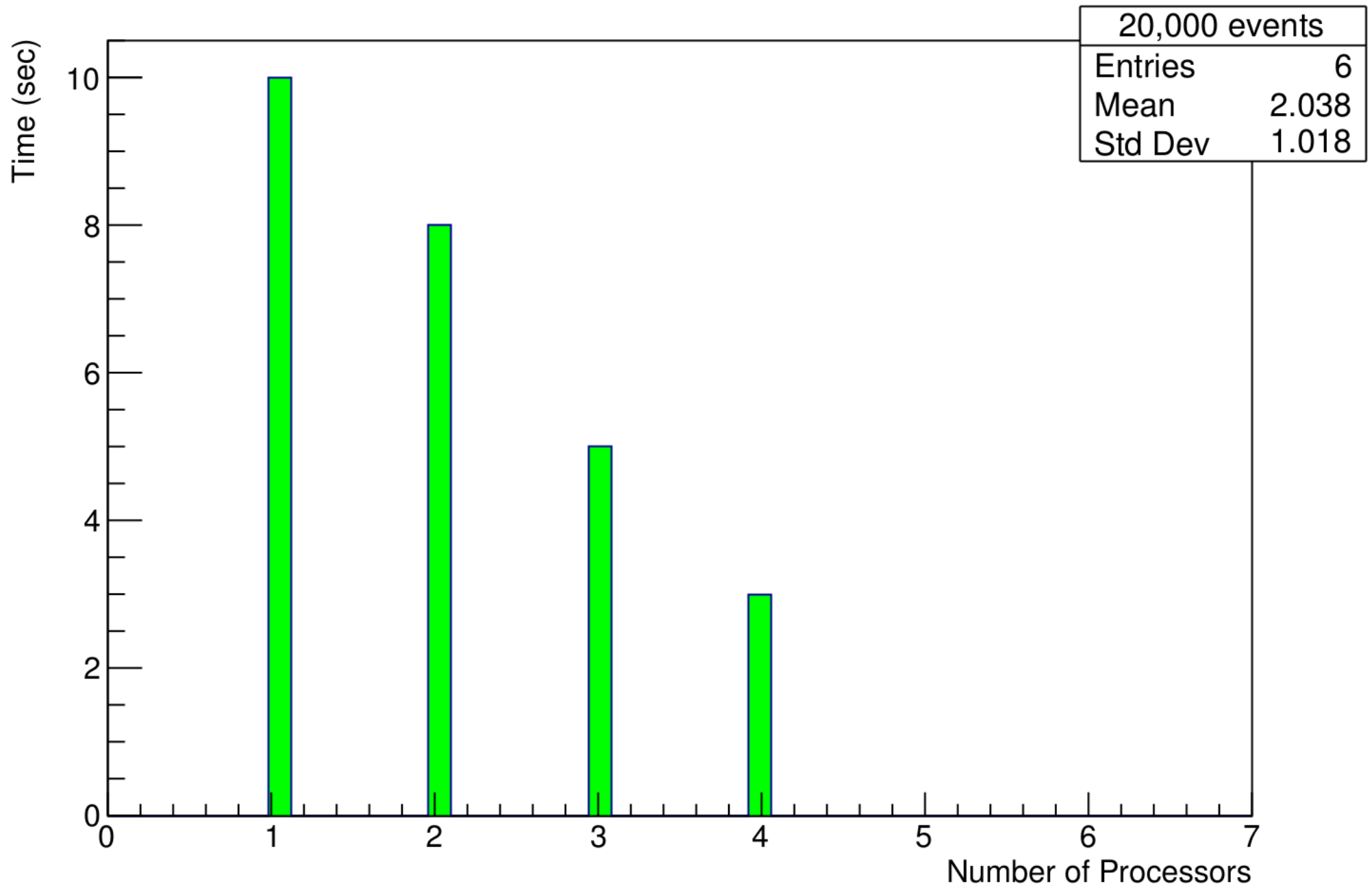
for(UInt_t k=0; k<ROCs.size(); ++k){
    sumFOM += ROCs.at(k);
}

if(!optParams){
    for(UInt_t l=0; l<ROCs.size(); ++l){
        if (rocIntegrals) rocIntegrals[l] = ROCs.at(l);
        std::cout << "Fold " << l+1 << " ROCIntegral: " << ROCs.at(l) << std::endl;
    }
    std::cout << "Average ROCIntegral: " << sumFOM/(double)NumFolds << std::endl;
}

return sumFOM/(double)NumFolds;

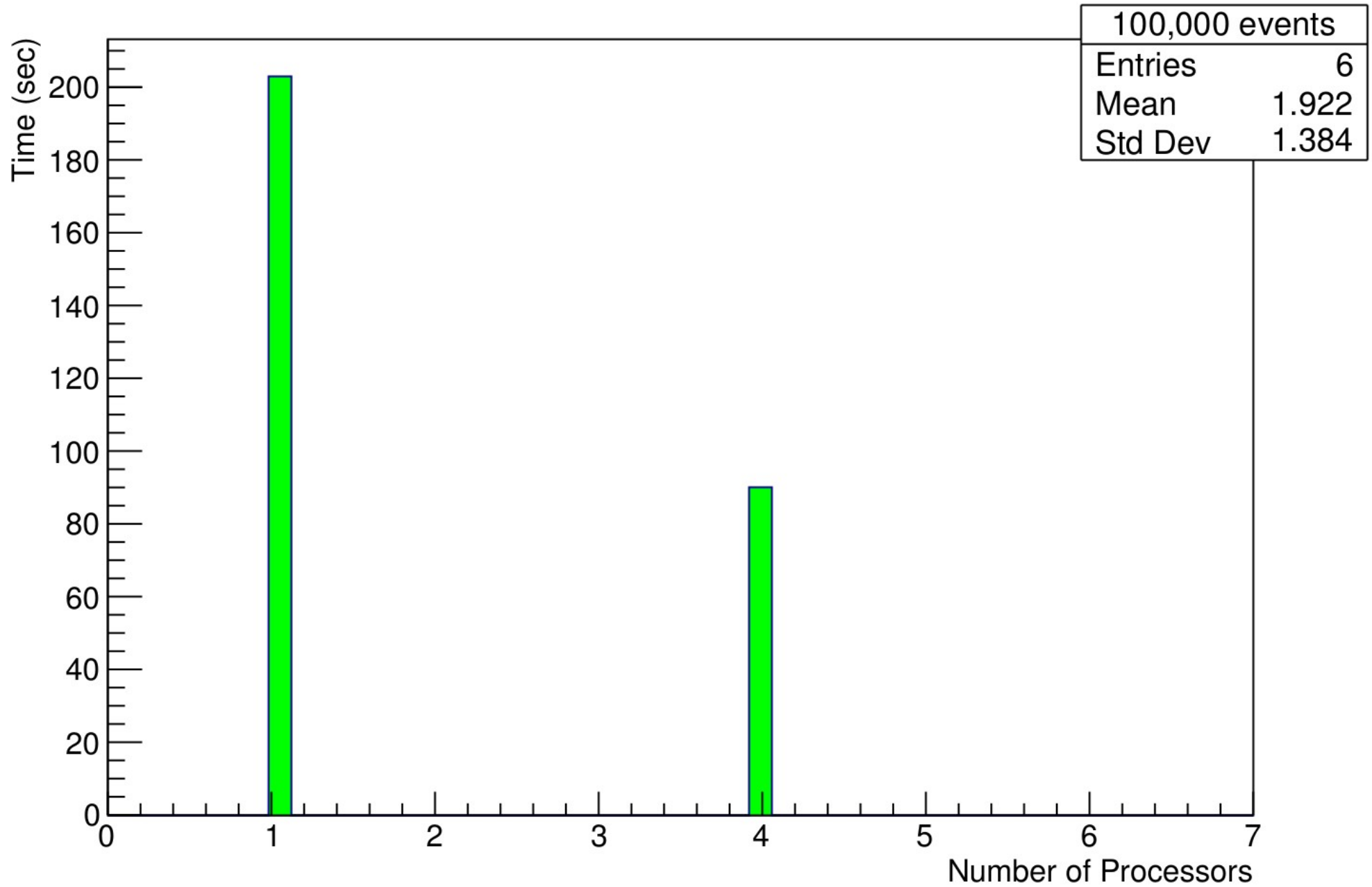
```

# 2 Folds



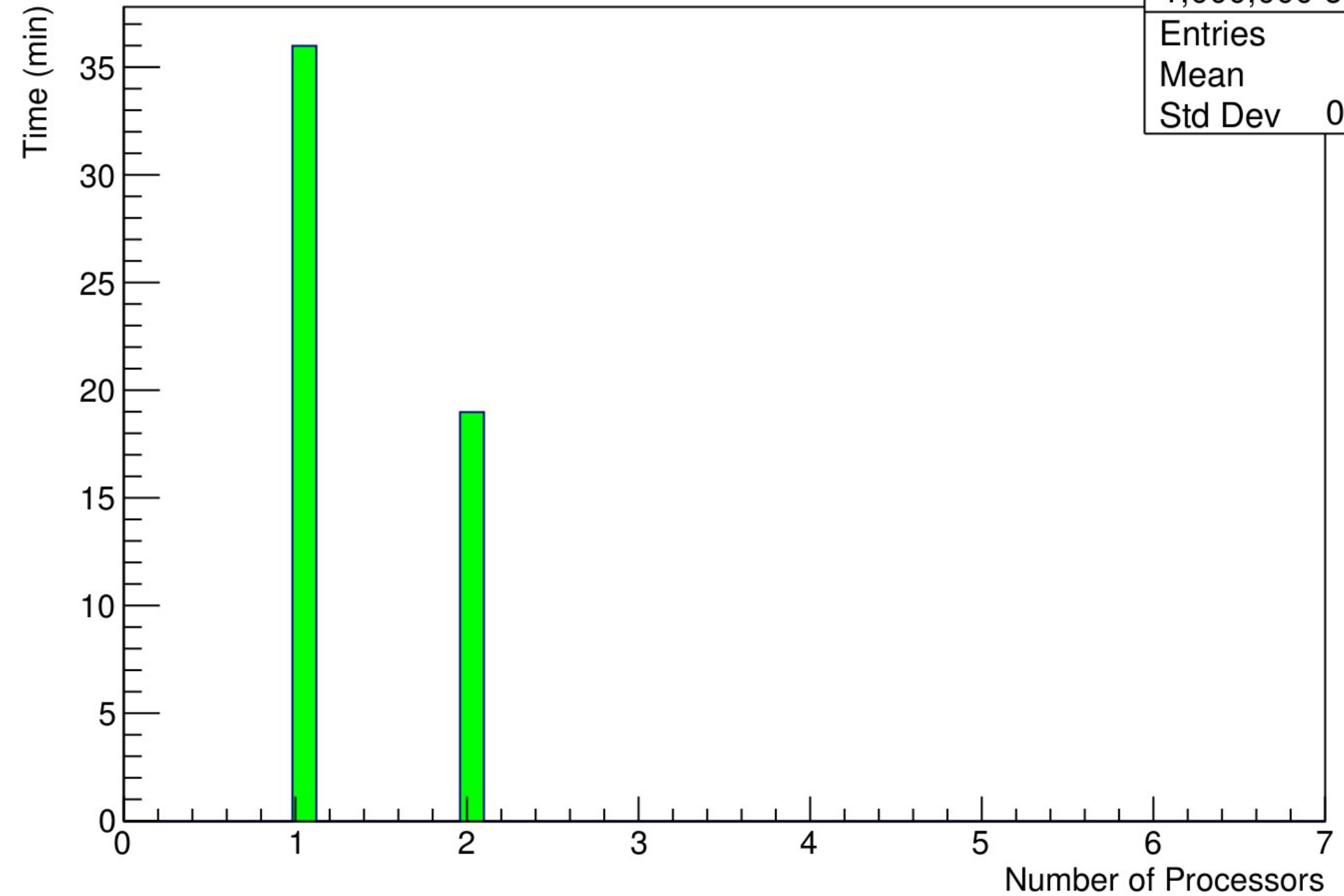


# 20 Folds



# 20 Folds

1,000,000 events	
Entries	6
Mean	1.345
Std Dev	0.4755



# Outlook

- Parallelization of BDT using Multithreading and Multiprocessing

Thank you very much for your attention!