The **Atom/FastLim** approach to Recasting

Ian-Woo Kim (CERN)
Kazuki Sakurai (King's College London)

in collaboration with **Michele Papucci** and **Andreas Weiler**

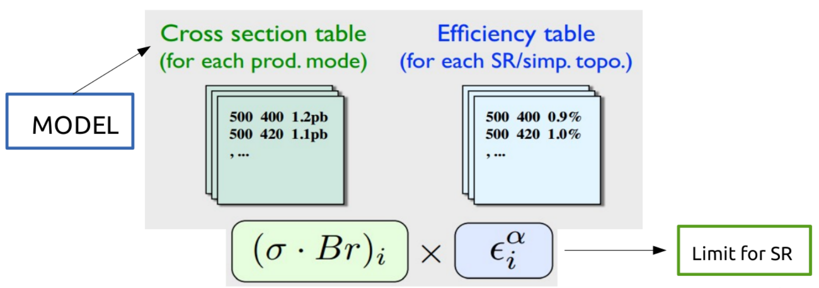
Mini-workshop on recasting ATLAS and CMS NP Searches

LPSC, Grenoble, France, 9 Sep 2014

Recasting ATLAS and CMS NP Searches

- **Recasting**: reinterprete ATLAS and CMS experimental results for specific limited models/parameters to a broader class of models/parameters
- We need two facilities for performing recasting:
  - Database for experimental analyses and results
  - Automated program pipeline of experimental analyses for different models
- Analyses are described best in terms of `code`!
- We need a *unified framework* for describing/performing/archiving analyses.

Factorize Recasting by Simplified Models



- Analyses for **Simplified Models** (or simplified topology) can be reused for different models

- We can factorize the task of recasting experimental limit by using Simplified Models.

$\rightarrow$ See Kazuki's presentation of `FastLim`.

Atom: Automated Tester Of Models

- `Rivet`-based Analysis Framework incorporating detector-level object reconstution and efficiency calculation
- Take MC events as input and then calculate efficiency tables and histograms defined in analyses
- Currently provides $\sim \mathcal{O}(100)$ Atom analyses (most of SUSY analyses and exotic searches) and compatible with `Rivet` analyses (mostly QCD validation analyses)
- Serve as an efficiency table calculation backend to `FastLim`

Atom idea: combine detector simulation and analysis

MC truth → Detector Simulation → Analysis

- Conventional separation (to reflect the real world situation)
  - **Detector Simulation**: From Monte Carlo truth, identify observable objects from the detector configurations and applying realistic effects.
  - **Analysis**: Using identified objects from detector simulation, apply some selections and/or extremize variables to construct a desirable test statistic for probabilistic interpretation

→ Practically, detector/analysis are not very well separated.

Atom idea: combine detector simulation and analysis

MC truth → Detector Simulation → Analysis

- Some objects are *not* really detector properties, but rather part of *analysis strategies*. *e.g.*
  - Various jet algorithms, Isolated objects
  - More nontrivial objects like b-tagged jets using complex neural network algorithms or MVA.
- There are limitations in implementing such complex objects for fast detector simulators, and this gives rise to need for tuning detector simulator. We have non-universal prolification of detector object definition.

Atom idea: combine detector simulation and analysis

Atom: MC truth $\longrightarrow$ Analysis(Detector)

- We scrap the idea of a separate fast detector simulation!
- Instead, every analysis starts from *the MC truth level*.
- Provide functions (called **projections**) to create more coarse-grained complex objects
- Detector simulation exists as projection libraries facilitating distribution distortion.

$\rightarrow$ need to model how to represent detector simulation in analysis

Modeling Distribution Distortion

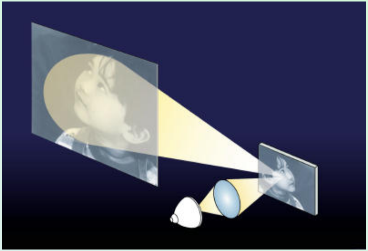
- Detector simulation $\rightarrow$ probability distribution distortion;
  Distortion can happen anywhere without actual association with detector components, *e.g.*
  - some not-well-modelled detector components
  - treating distribution of reconstructed objects (such as top-quark/tagged jet object) as distorted from the truth for convenience with limited scope
- In general, distortion can happen at any step.
  Basic object identification, isolated objects, reconstructed objects

Modeling Distribution Distortion

- At 1st order, we can describe the distortion in terms of Efficiency/Rejection/Smearing.
  - Assumption: the inner-correlation between measured object quantities are well cancelled out. (assumption of atomicity)
  - Efficiency: matched probability P for object identification hypothesis
  - Rejection: mismatched probability $\widetilde{P}$ for object null-identification hypothesis
  - Smearing: bin-to-bin migration W_{ij} for a given distribution

$\rightarrow$ In `Atom`, every final-state **projection** can be specified with associated Efficiency/Rejection/Smearing parameters

Rivet::Projection

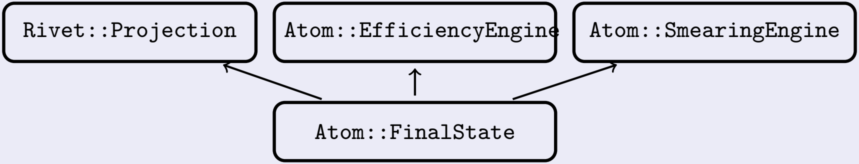


- A `Rivet::Projection` is a registered function extracting a quantity out of an event, the result of which can be cached in memory.
- In `Rivet`, every quantity extracted from events should be through `Projection`s.
- Analysis = a collection of projections / histogram filling from the result of projection with certain event-by-event criteria.
- `Projection` is identified with its name and also the defining data. `Rivet::ProjectionHandler` keeps a list of registered projections.

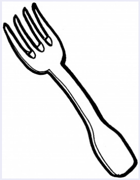
BTW, it's not a projection in mathematical sense! ($P^2 \neq P$)

Rivet::FinalState → Atom::FinalState

- **Rivet::FinalState** is a base projection for all the final state objects from an MC event. This is effectively the mother of all the useful projections.
- **Atom::FinalState** has Efficiency/Smearing Engine in addition to **Rivet::FinalState**.



- Due to the design changes including this, **Atom** is currently a fork of **Rivet**.



List of Projections used in `Atom`

- From `Rivet` with new `Atom::FinalState`:

```
FinalState, ChargedFinalState, FastJets,
IdentifiedFinalState, LeptonClusters, MergedFinalState,
MissingMomentum, VetoedFinalState, VisibleFinalState
```

- `Atom` only:

```
IsolatedParticle,IsoElectron, IsoMuon, IsoPhoton, IsoTrack,
NearIsoParticle, PhotonClusters, JetFinalState, JetTagger,
TauFinder, HTTools, HardProcess, InvMassFinalState, AlphaT,
Razor
```

- The list is growing as new projections are needed in the analyses. Especially, new isolation strategies are continuosly being added as the ATLAS and CMS analyses evolve.

Atom Output: Histogram/Efficiency table

- Outputs of Rivet analyses are defined as YODA histogram.
  - Histogram is booked in the initialization stage.
  - Histogram binning is matched with predefined histogram information externally registered as separate files $\rightarrow$ good for validation

- Atom shares Rivet's histogramming functionality and also provides cut-and-count efficiency tables.
  - Efficiency table is booked in the initialization stage.
  - Atom automatically classify subprocesses and calculate efficiencies for booked efficiency tables
  - Output format is YAML which is both human- and machine-readable.

Atom Output: Histogram/Efficiency table

- Efficiency Table creation example:

Analysis

```
class ATLAS_2013_CONF_2013_061 : .. {
  // Analysis initialization
  void initLocal() {
    // ...
    // register efficiency table
    _effh.bookEfficiency("eff");
    _effh.bookEfficiency("eff2");
    // ...
  }

  // for each event
  void analyzeLocal(..) {
    // ...
    _effh.PassEvent("eff");
    // ...
    _effh.PassEvent("eff2");
  }
  ..
}
```

Output

```
Efficiencies:
  - ATLAS_2013_CONF_2013_061:
      - Sub-process ID: 0
        Efficiency Name: eff
        Efficiency Value: 0.90
        Efficiency Value Stat Error:
          [0.006,0.006]
      - Sub-process ID: 0
        Efficiency Name: eff2
        Efficiency Value: 0.264
        Efficiency Value Stat Error:
          [0.004,0.004]
      ...
```

Efficiency table creation is very simple!

Atom Input: Efficiency/Rejection/Smearing

- One can define a collection of Efficiency/Rejection/Smearing tables for some common detector objects such as electrons or jets as YAML objects.
- Stored as separate files in $ATOM_YAML_PATH environment variables. Not mixed with source codes. → Easy to configure, test and track them!
- Can import files and combine multiple files into one single collections.

Atom Input: Efficiency/Rejection/Smearing

- Sample YAML file for a collection of efficiency/smearing tables

ATLAS2011.yaml


```yaml
Name: ATLAS2011
Class: TopLevel
Description: ATLAS 2011 detector
Reference: arXiv:xxxx.yyyy
Comment: ...
ValidationInfo: Validated on 2014/02
Range:
    - Import: Range_Full_ATLAS
Identification:
    Electron:
        - Import: Electron_Tight_ATLAS
        - Import: Electron_Medium_ATLAS
        - Import: Electron_Loose_ATLAS
    Photon: ...
    Jet: ...
    ...
Smearing:
    ...
    Jet:
        - Import: Smear_TopoJet_ATLAS
    ...
```


Electron_Tight_ATLAS.yaml


```yaml
Name: Electron_Tight_ATLAS
Tag: ATLAS
Description: Tight electron object 2011 ATLAS
Comment: We use table from reference
Reference: arXiv:xxxx.yyyy
Efficiency:
    Type: Grid
    PtBins: [4.0,7.0,10.0,15.0,20.0,..]
    EtaBins: [-2.5,-2.0,-1.52,-1.37,..]
    IsEtaSymmetric: False
    Grid:
        Type: Full
        Data:
            [ [ 0.75,0.75,0.75,0.75,0.75,..]
            , [ 0.75,0.75,0.74,0.73,0.73,..]
            , [ 0.75,0.75,0.74,0.73,0.73,..]
            ...
```

Atom Input: Efficiency/Rejection/Smearing

- Sample YAML file for a collection of efficiency/smearing tables

ATLAS2011.yaml


```yaml
Name: ATLAS2011
Class: TopLevel
Description: ATLAS 2011 detector
Reference: arXiv:xxxx.yyyy
Comment: ...
ValidationInfo: Validated on 2014/02
Range:
    - Import: Range_Full_ATLAS
Identification:
    Electron:
        - Import: Electron_Tight_ATLAS
        - Import: Electron_Medium_ATLAS
        - Import: Electron_Loose_ATLAS
    Photon: ...
    Jet: ...
    ...
Smearing:
    ...
    Jet:
        - Import: Smear_TopoJet_ATLAS
    ...
```


Smear_TopoJet_ATLAS.yaml


```yaml
Name: Smear_TopoJet_ATLAS
Tag: ATLAS
Description: topojet
Comment: table
Reference: XXX
Smearing:
    Type: Interpolation
    IsEtaSymmetric: True
    Interpolation:
        Type: PredefinedMode3
        EtaBound: 4.0
        EtaBinContent:
            - BinStart: 0.0
              BinContent:
                  [ [ -2, 9.476216187754203 ]
                  , [ -1, -0.1693988048822812 ]
                  ...
            - BinStart: 0.75
              BinContent:
                  [ [ -2, 8.197400117302609 ]
                  , [ -1, -5.636233086517818e-2 ]
                  ...
```


Analytic interpolation: $\delta p_T / p_T = \sum_i c_i |p_T|^{\alpha_i}$

Atom Input: Efficiency/Rejection/Smearing

- Use Efficiency/Smearing table in an analysis

Analysis

```
class ATLAS_2013_CONF_2013_061 : .. {
private:
  detector_parameter_t dp;
public:
  // Analysis initialization
  void initLocal() {
    dp = findDetector("ATLAS2011");
    RangeSelector testRange = dp.range("Range_Full_ATLAS");
    ...
    IsoElectron ele = IsoElectron(testRange);
    ele.setSmearing( Selected, dp.electronSim("Smear_Electron_ATLAS");
    ele.setEfficiency( Selected, dp.electronEff("Electron_Tight_ATLAS");
    addProjection( ele, "Electrons" );
    // ...
  }

  // for each event
  bool analyzeLocal(const Event& ev, ..) {
     const Particles eles =
       applyProjection<IsoElectron>(event,"Electrons").particlesByPt();
    ...
    for( unsigned int i = 0 ; i < eles.size(); i++ ) {
    ...
  }
  ..
}
```

Atom interface

- Atom provides both a command-line interface and a python shell interpreter interface.
  - Command-line interface: one can launch analyses with command-line flags like

```
$ atom -a ATLAS_2013_CONF_2013_061 -H test.hepmc -n 1000
```

  - Python shell interface: one can use atom in interpreter mode.

```
$ atom
Atom 0.9 running on machine pcth251.cern.ch (x86_64)
atom> show Analyses
ATLAS_2010_S8755477       ATLAS_2010_S8814007
..
atom> addAnalyses ATLAS_2010_S8755477
Atom.AnalysisHandler: INFO  register Analysis: ATLAS_2010_S8755477
atom> addInput test.hepMC HepMC
Atom.ReaderHandler: INFO  register Reader: HepMC
atom> launch
```

  - Interpreted commands can be collected and run as a single batch file.

```
$ atom -b batchfile
```

Atom interface

- Atom can support various event file formats:
  - HepMC, LHE and StdHep

- multiple inputs and analyses and run at once.
  - Rivet projections are cached, so duplicated calculations are efficiently handled.

- Analyses are registered with meta-informations

```
atom> show Analysis ATLAS_2010_S8755477
Analysis:    ATLAS_2010_S8755477
Description:
Dijet mass spectrum from 0.2~TeV to 1.8~TeV in $pp$ collisions at $\sqrt{s} =
  7$ TeV, based on an integrated luminosity of 315 nb$^{-1}$.

Abstract: ...
Collider: LHC 7 TeV
Run:
Experiment: ATLAS
Year:       2010
Identifiers: <arXiv:1008.2461> [bibTeX] ...
Status:     VALIDATED
```

Currently, most of the meta-informations are missing although analyses are ready. We keep adding contents.

Currently implemented analyses: ATLAS I

```
ATLAS_2010_S8755477      : 2-jet                      @ 315 nb^-1, 7 TeV
ATLAS_2010_S8814007      : 2-jet angular              @ 3.1 pb^-1, 7 TeV
ATLAS_2010_S8914249      : 2-photon + MET             @ 7 TeV
ATLAS_2011_CONF_2011_036 : ttbar+MET                  @ 35 pb^-1,   7 TeV
ATLAS_2011_CONF_2011_039 : SUSY multilepton+jets+MET  @ 35 pb^-1,   7 TeV
ATLAS_2011_CONF_2011_086 : SUSY jets+MET              @ 165 pb^-1,  7 TeV
ATLAS_2011_CONF_2011_090 : SUSY 1lep+jets+MET         @ 165 pb^-1,  7 TeV
ATLAS_2011_CONF_2011_096 : monojet + MET              @ 1 fb^-1,    7 TeV
ATLAS_2011_CONF_2011_098 : SUSY 0lep+b-jets+MET       @ 0.83 fb^-1, 7 TeV
ATLAS_2011_CONF_2011_123 : ttbar resonance dilep      @ 1.04 fb^-1, 7 TeV
ATLAS_2011_CONF_2011_126 : like-sign muon pair        @ 1.6 fb^-1,  7 TeV
ATLAS_2011_CONF_2011_130 : SUSY 1lep+b-jets+MET       @ 1 fb^-1,    7 TeV
ATLAS_2011_CONF_2011_144 : 4 charged lep              @ 1.02 fb^-1, 7 TeV
ATLAS_2011_S8970084      : SUSY 1lep+jets+MET         @ 35 pb^-1,   7 TeV
ATLAS_2011_S8983313      : SUSY jets+MET              @ 35 pb^-1,   7 TeV
ATLAS_2011_S8996709
ATLAS_2011_S9011218      : SUSY b-jets+MET            @ 35 pb^-1,   7 TeV
ATLAS_2011_S9019553      : SUSY same flavor leps      @ 35 pb^-1,   7 TeV
ATLAS_2011_S9019561      : SUSY 2lep+MET              @ 35 pb^-1,   7 TeV
ATLAS_2011_S9108483      :
ATLAS_2011_S9120726      : diphoton + MET             @ 36 pb^-1,   7 TeV
ATLAS_2011_S9203559      :
ATLAS_2011_S9225137      : large jets + MET           @ 1.34 fb^-1, 7 TeV
ATLAS_2012_CONF_2012_033 : 0lep squark,gluino         @ 7 TeV
ATLAS_2012_CONF_2012_084 : dark matter with jet+MET   @ 4.7 fb^-1,  7 TeV
ATLAS_2012_CONF_2012_088 : dijet mass                 @ 5.8 fb^-1,  8 TeV
ATLAS_2012_CONF_2012_109 : SUSY jets+MET              @ 5.8 fb^-1,  8 TeV
ATLAS_2012_CONF_2012_147 : monojet + MET              @ 10 fb^-1,   8 TeV
ATLAS_2012_CONF_2012_148 : dijet mass                 @ 13 fb^-1,   8 TeV
ATLAS_2012_I1189659      : dijet mass                 @ 7 TeV
ATLAS_2012_I1204447      : 3lep                       @ 7 TeV
```

Currently implemented analyses: ATLAS II

```
ATLAS_2013_CONF_2013_007 : SUSY same sign leps       @ 21 fb^-1,   8 TeV
ATLAS_2013_CONF_2013_024 : top squark had ttbar+MET  @ 21 fb^-1,   8 TeV
ATLAS_2013_CONF_2013_035 : SUSY neut prod w/ 3lep+MET@ 21 fb^-1,   8 TeV
ATLAS_2013_CONF_2013_037 : SUSY stop w/ 1lep+jet+MET @ 21 fb^-1,   8 TeV
ATLAS_2013_CONF_2013_047 : SUSY jet+MET              @ 20.3 fb^-1, 8 TeV
ATLAS_2013_CONF_2013_048 : SUSY stop w/ 2lep + MET   @ 20 fb^-1,   8 TeV
ATLAS_2013_CONF_2013_049 : SUSY slep 0jet+2opp lep+MET@ 20 fb^-1,8 TeV
ATLAS_2013_CONF_2013_053 : SUSY 3rd 2bjet+MET        @ 20.1 fb^-1, 8 TeV
ATLAS_2013_CONF_2013_054 : large jet+MET             @ 20 fb^-1,   8 TeV
ATLAS_2013_CONF_2013_061 : SUSY 3-bjet+MET           @ 20.1 fb^-1, 8 TeV
ATLAS_2013_CONF_2013_068 : SUSY stop->c neut         @ 20.3 fb^-1, 8 TeV
ATLAS_2013_CONF_2013_093 : SUSY neut, 1lep+1(h->bb)+MET @ 20.3 fb^-1, 8 TeV
ATLAS_2014_I1286444      : SUSY 2lep+(b)jet+MET      @ 20.3 fb^-1, 8 TeV
ATLAS_2014_I1286761      : SUSY 2lep+MET             @ 20.3 fb^-1, 8 TeV
```

Currently implemented analyses: CMS

```
CMS_2010_S8820767   : dijet resonance  @ 2.9 pb^-1,  7 TeV
CMS_2011_I919742    : dijet mass resonance @ 1 fb^-1, 7 TeV
CMS_2011_S8932190   : SUSY (>=2) jets+MET @ 35 pb^-1, 7 TeV
CMS_2011_S8990433   : SUSY diphoton+MET @ 35 pb^-1, 7 TeV
CMS_2011_S8991847   : SUSY opp 2lep      @ 35 pb^-1, 7 TeV
CMS_2011_S9036504   : SUSY same 2lep     @ 35 pb^-1, 7 TeV
CMS_2012_I1090423   : quark composite in dijet angular @ 2.2 fb^-1, 7 TeV
CMS_2012_I1119567   : heavy top, 2lep   @ 1.14 fb^-1,  7 TeV
CMS_2012_I1189823   : quantum black hole, b-dijet @ 7 TeV
CMS_2013_I1215599   : dijet                        @ 7 TeV
CMS_2013_I1220378   : contact int, jet PT          @ 5.0 fb^-1,  7 TeV
CMS_PAS_EXO_11_036  : heavy bottom                 @ 1.14 fb^-1, 7 TeV
CMS_PAS_EXO_11_050  : heavy top in dilep           @ 1.14 fb^-1, 7 TeV
CMS_PAS_EXO_11_051  : top prime pair in lep+jets   @ 1.4 fb^-1,  7 TeV
CMS_PAS_EXO_11_059  : Monojet+MET                  @ 1.1 fb^1,   7 TeV
CMS_PAS_EXO_12_048  : Monojet+MET                  @ 19.5 fb^-1, 8 TeV
CMS_PAS_EXO_12_059  : dijet mass                   @ 19.6 fb^-1, 8 TeV
CMS_PAS_SUS_10_005  : SUSY jets+MET                @ 36 pb^-1,   7 TeV
CMS_PAS_SUS_10_009  : SUSY squark+gluino inclusive@ 35 pb^-1,   7 TeV
CMS_PAS_SUS_10_011  : SUSY dijet, multijet+ MET    @ 35 pb^-1,   7 TeV
CMS_PAS_SUS_11_003  : SUSY jets + MET              @ 1.14 fb^-1, 7 TeV
CMS_PAS_SUS_11_004  : SUSY had jets + MET          @ 1.1 fb^-1,  7 TeV
CMS_PAS_SUS_11_005  : SUSY had jets + MT2          @ 1.1 fb^-1,  7 TeV
CMS_PAS_SUS_11_006  : SUSY b-jets + MET            @ 1.1 fb^-1,  7 TeV
CMS_PAS_SUS_11_010  : SUSY same 2lep+jet+MET       @ 0.98 fb^-1, 7 TeV
CMS_PAS_SUS_11_011  : SUSY opp 2lep+MET            @ 0.98 fb^-1, 7 TeV
CMS_PAS_SUS_11_015  : SUSY lep+MET                 @ 1.1 fb^-1,  7 TeV
CMS_PAS_SUS_11_017  : SUSY Z+MET                   @ 0.98 fb^-1, 7 TeV
CMS_PAS_SUS_11_022  : SUSY 0->3bjet+MET            @ 4.98 fb^-1, 7 TeV
CMS_PAS_SUS_11_028  : SUSY 1lep+bjet+MET           @ 4.98 fb^-1, 7 TeV
CMS_PAS_SUS_12_011  : SUSY multijet+MET            @ 4.98 fb^-1, 7 TeV
CMS_PAS_SUS_12_028  : SUSY 0->4bjet+MET            @ 11.7 fb^-1, 8 TeV
CMS_PAS_SUS_13_012  : SUSY multijet+MET            @ 19.5 fb^-1, 8 TeV
CMS_PAS_TOP_11_005  : TOP ttbar xsec in 2lep       @ 2.3 fb^-1,  7 TeV
```

Conclusion

- Recasting ATLAS and CMS New Physics Searches require a new tool which can serve as a unified framework for describing/performing/archiving analyses.
- `Atom` is `Rivet`-based analysis framework incorporating detector simulation and efficiency calculating
- We extend `Rivet::Projection` with efficiency/rejection/smearing functionality.
- `Atom` user can generate efficiency table very easily due to `Atom`'s automatic `YAML` output.
- Predefined collection of efficiency/smearing can be easily added by creating `YAML` object in separate files. One can use it when defining and registering a new projection.
- Atom provides a flexible method for running analyses both in