RECAST

Lukas Heinrich

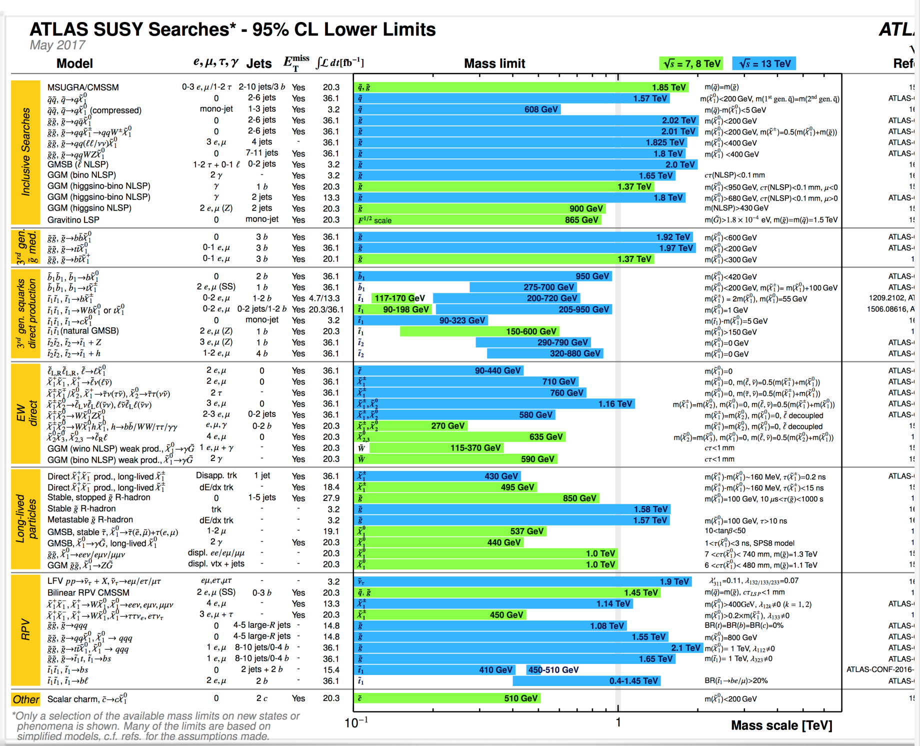
US ATLAS Workshop 2017
Open Session

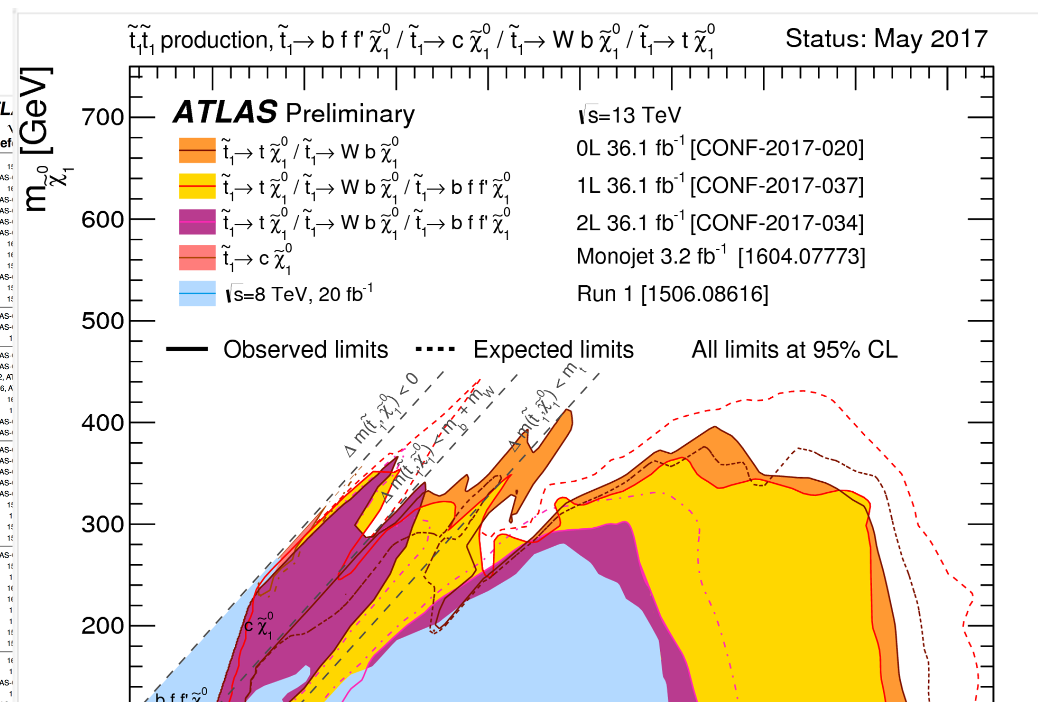
NEW YORK UNIVERSITY

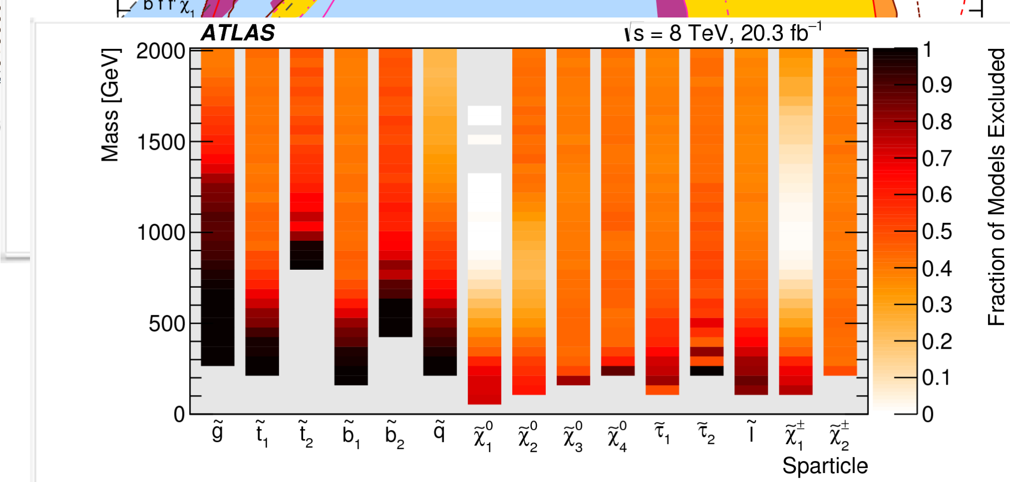
the need for reinterpretation

After the Higgs discovery completed the Standard Model, the search for BSM physics has become an even higher priority.

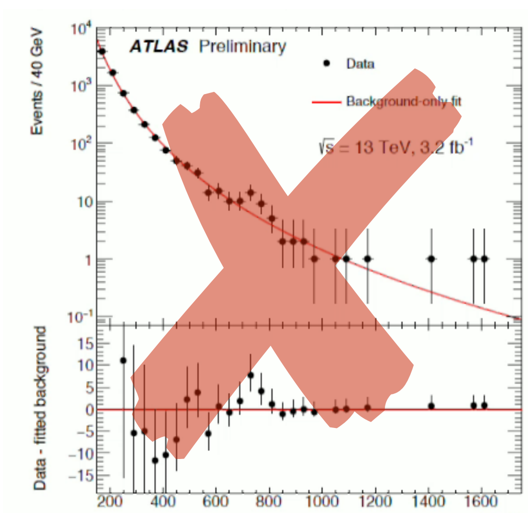
ATLAS is producing tons of results.. so far we have not found any significant excess (sorry no 750 GeV…)







average sad theorist



the need for reinterpretation

Where is the New Physics?

- hide in unexpected places, complex final states, low-rate / low-acceptance scenarios (e.g. compressed models, models spreading across many topologies)
- not be reachable at all at the LHC

how do we exploit the LHC data such to maximize our understanding of still viable models?

Problem:

there are ***many more*** candidate models than we have graduate students to design dedicated analyses for each new model — let's make the most of the analyses that we have. **Many of them are sensitive to a whole range of models.**

the need for reinterpretation

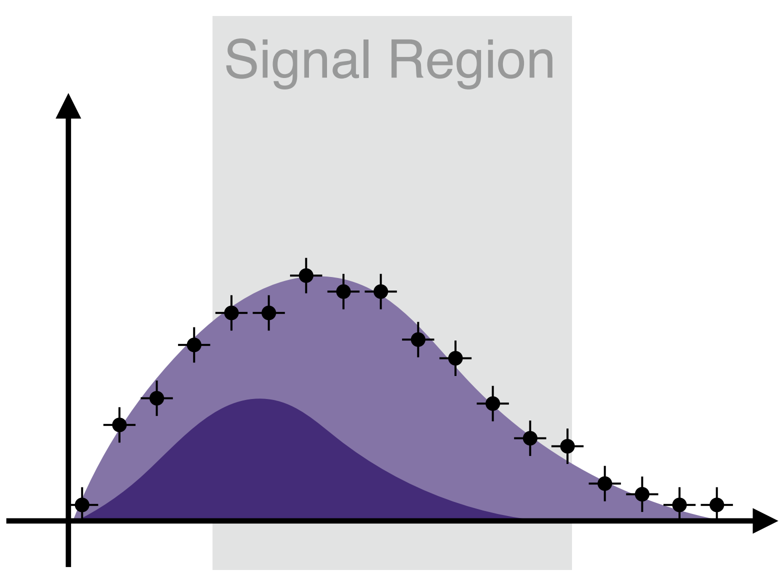
The analyses we prepare at the LHC are **high-effort, expensive projects**: non-trivial amount of person-power, time, and computing resources devoted to achieving a publication-quality result.

Most of the work goes into: **taking data, designing**, **validating** the analysis strategy, **understanding Standard Model backgrounds.** Effectively: a measurement of observed and backgrounds in interesting phase space regions.

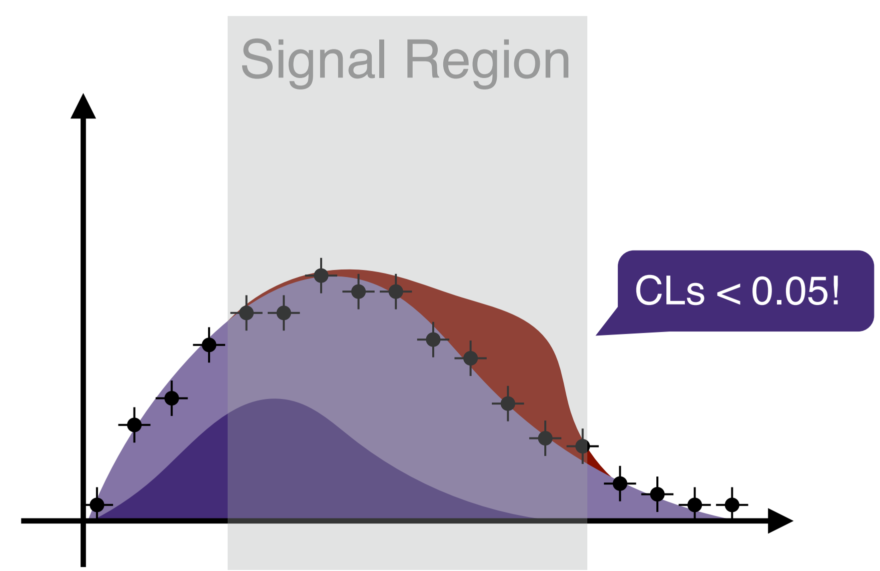
Model interpretation come at the end, and are technically the **easiest part:** analysis pipeline is **fixed** after unblinding, MC dataset sizes small. Analysis teams routinely check hundreds of parameter points (of their favorite model).

But: most analyses only **interpreted once** within limited set of models.

- analysis team pushing for conference deadline
- interesting models proposed by hep-ph *after* they've seen the paper / note.



observed **data** and **estimated backgrounds**



interpreted w.r.t **model A**

the need for reinterpretation

The analyses we prepare at the LHC are **high-effort, expensive projects**: non-trivial amount of person-power, time, and computing resources devoted to achieving a publication-quality result.

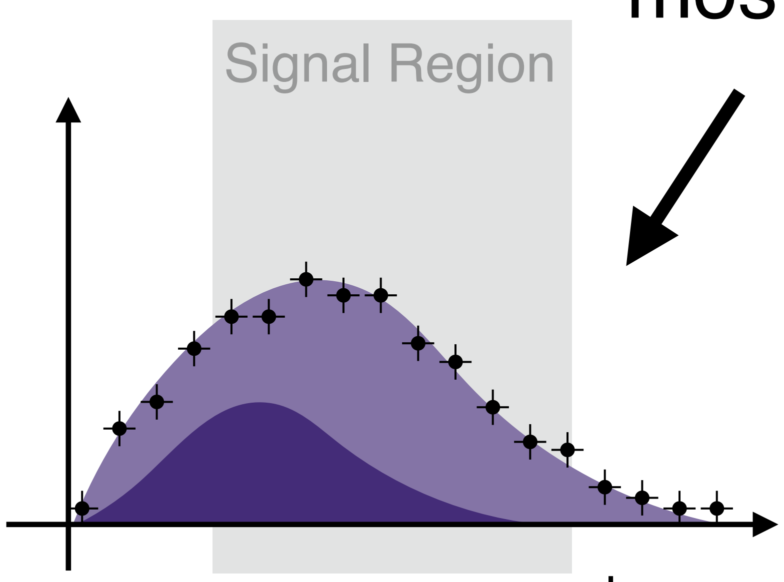
Most of the work goes into: **taking data, designing**, **validating** the analysis strategy, **understanding Standard Model backgrounds.** Effectively: a measurement of observed and backgrounds in interesting phase space regions.

Model interpretation come at the end, and are technically the **easiest part:** analysis pipeline is **fixed** after unblinding, MC dataset sizes small. Analysis teams routinely check hundreds of parameter points (of their favorite model).

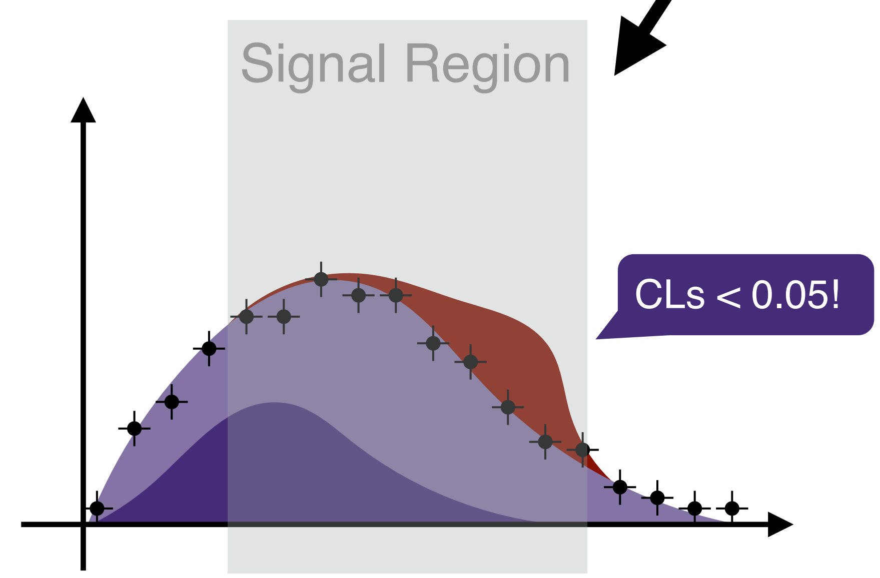
But: most analyses only **interpreted once** within limited set of models.
- analysis team pushing for conference deadline
- interesting models proposed by hep-ph *after* they've seen the paper / note.

~ easy

most of the work!



observed **data** and
estimated backgrounds



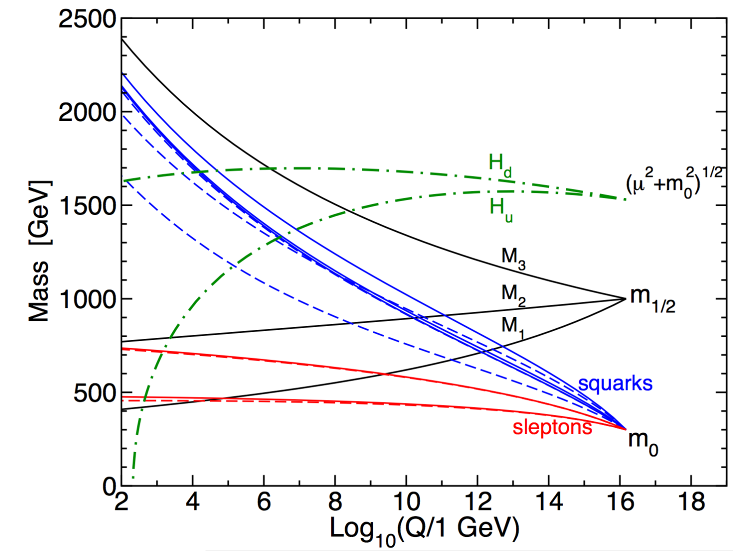
interpreted w.r.t **model A**

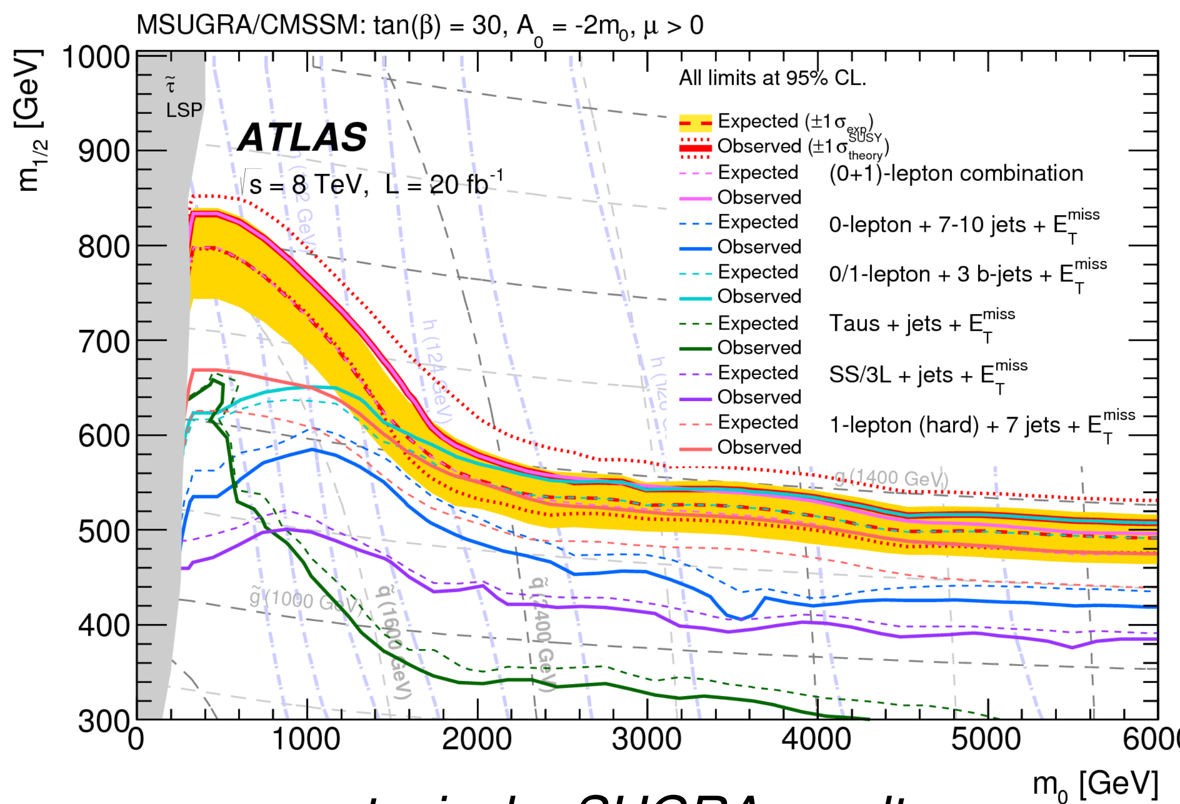
enabling reinterpretation

MSSM has 120 parameters — need to cut down drastically to evaluate models.

Early in the LHC era: UV-inspired models with very harsh symmetry constraints on MSSM parameters:

- ⊕ low dimensionality, easy interpretability as a full theory
- ⊖ rigid relationships of parameters not necessarily realistic
- ⊖ hard to *reinterpret (what does an excluded cMSSM point tell us about other models?)*



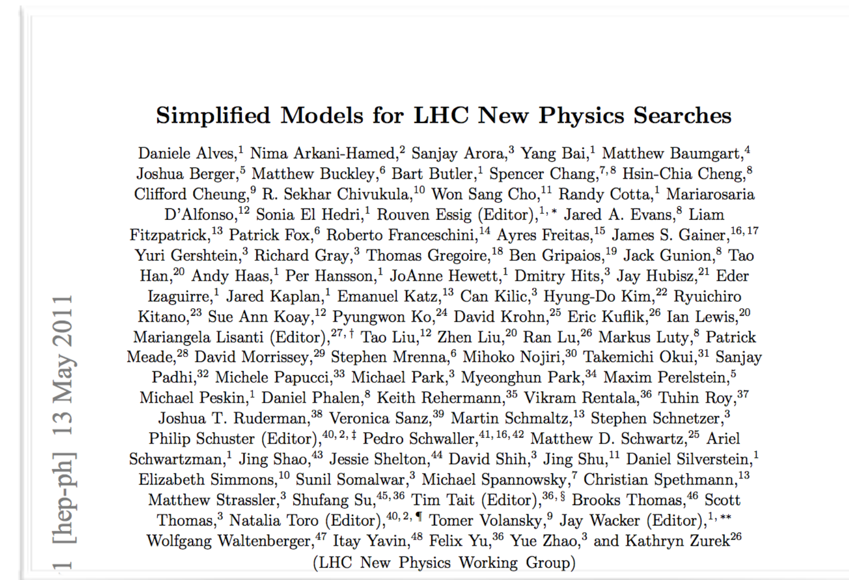


typical mSUGRA result

enabling reinterpretation

The move to simplified models: standard SUSY (and increasingly non-prompt + non-SUSY) searches moved to setting limits on *simplified models acting as surrogate for model class*

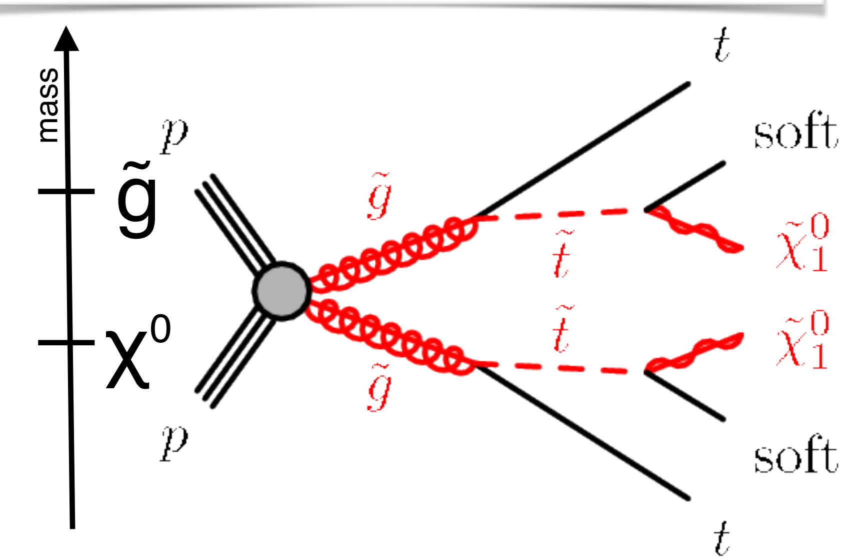
- ⊕ focus on decay chains to which LHC is sensitive, easier to reinterpret.
- ⊖ not a full SUSY model. Reinterpretation is mandatory.



1 [hep-ph] 13 May 2011

Simplified Models for LHC New Physics Searches

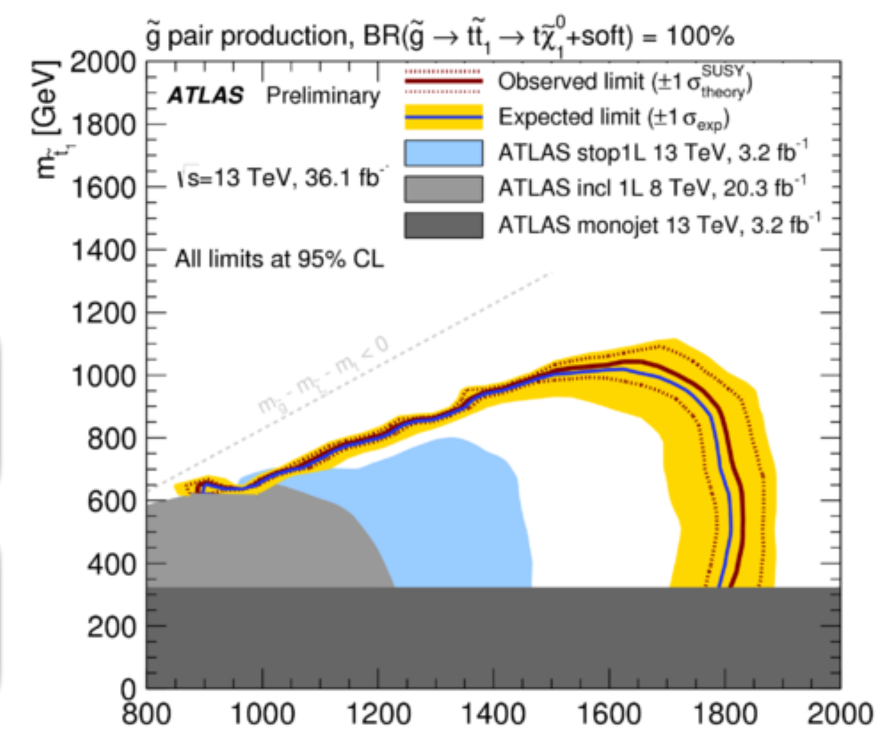
Daniele Alves,[1] Nima Arkani-Hamed,[2] Sanjay Arora,[3] Yang Bai,[1] Matthew Baumgart,[4] Joshua Berger,[5] Matthew Buckley,[6] Bart Butler,[1] Spencer Chang,[7,8] Hsin-Chia Cheng,[8] Clifford Cheung,[9] R. Sekhar Chivukula,[10] Won Sang Cho,[11] Randy Cotta,[1] Mariarosaria D'Alfonso,[12] Sonia El Hedri,[1] Rouven Essig (Editor),[1,*] Jared A. Evans,[8] Liam Fitzpatrick,[13] Patrick Fox,[6] Roberto Franceschini,[14] Ayres Freitas,[15] James S. Gainer,[16,17] Yuri Gershtein,[3] Richard Gray,[3] Thomas Gregoire,[18] Ben Gripaios,[19] Jack Gunion,[8] Tao Han,[20] Andy Haas,[1] Per Hansson,[1] JoAnne Hewett,[1] Dmitry Hits,[3] Jay Hubisz,[21] Eder Izaguirre,[1] Jared Kaplan,[1] Emanuel Katz,[13] Can Kilic,[3] Hyung-Do Kim,[22] Ryuichiro Kitano,[23] Sue Ann Koay,[12] Pyungwon Ko,[24] David Krohn,[25] Eric Kuflik,[26] Ian Lewis,[20] Mariangela Lisanti (Editor),[27,†] Tao Liu,[12] Zhen Liu,[20] Ran Lu,[26] Markus Luty,[8] Patrick Meade,[28] David Morrissey,[29] Stephen Mrenna,[6] Mihoko Nojiri,[30] Takemichi Okui,[31] Sanjay Padhi,[32] Michele Papucci,[33] Michael Park,[3] Myeonghun Park,[34] Maxim Perelstein,[5] Michael Peskin,[1] Daniel Phalen,[8] Keith Rehermann,[35] Vikram Rentala,[36] Tuhin Roy,[37] Joshua T. Ruderman,[38] Veronica Sanz,[39] Martin Schmaltz,[13] Stephen Schnetzer,[3] Philip Schuster (Editor),[40,2,‡] Pedro Schwaller,[41,16,42] Matthew D. Schwartz,[25] Ariel Schwartzman,[1] Jing Shao,[43] Jessie Shelton,[44] David Shih,[3] Jing Shu,[11] Daniel Silverstein,[1] Elizabeth Simmons,[10] Sunil Somalwar,[3] Michael Spannowsky,[7] Christian Spethmann,[13] Matthew Strassler,[3] Shufang Su,[45,36] Tim Tait (Editor),[36,§] Brooks Thomas,[46] Scott Thomas,[3] Natalia Toro (Editor),[40,2,¶] Tomer Volansky,[9] Jay Wacker (Editor),[1,**] Wolfgang Waltenberger,[47] Itay Yavin,[48] Felix Yu,[36] Yue Zhao,[3] and Kathryn Zurek[26]
(LHC New Physics Working Group)



reinterpretation: calculate your models cross-section into simplified topology, compare to cross section limit. unlocked access to reinterpretation of many more models with a single analysis

Caveats:

- only works for models with same topology (only change the rates via xsec, signal shape stays static)
- complex models may have very low BR into ***any one simplified topology***.



strong limit on simplified model ≠ strong limit on real model

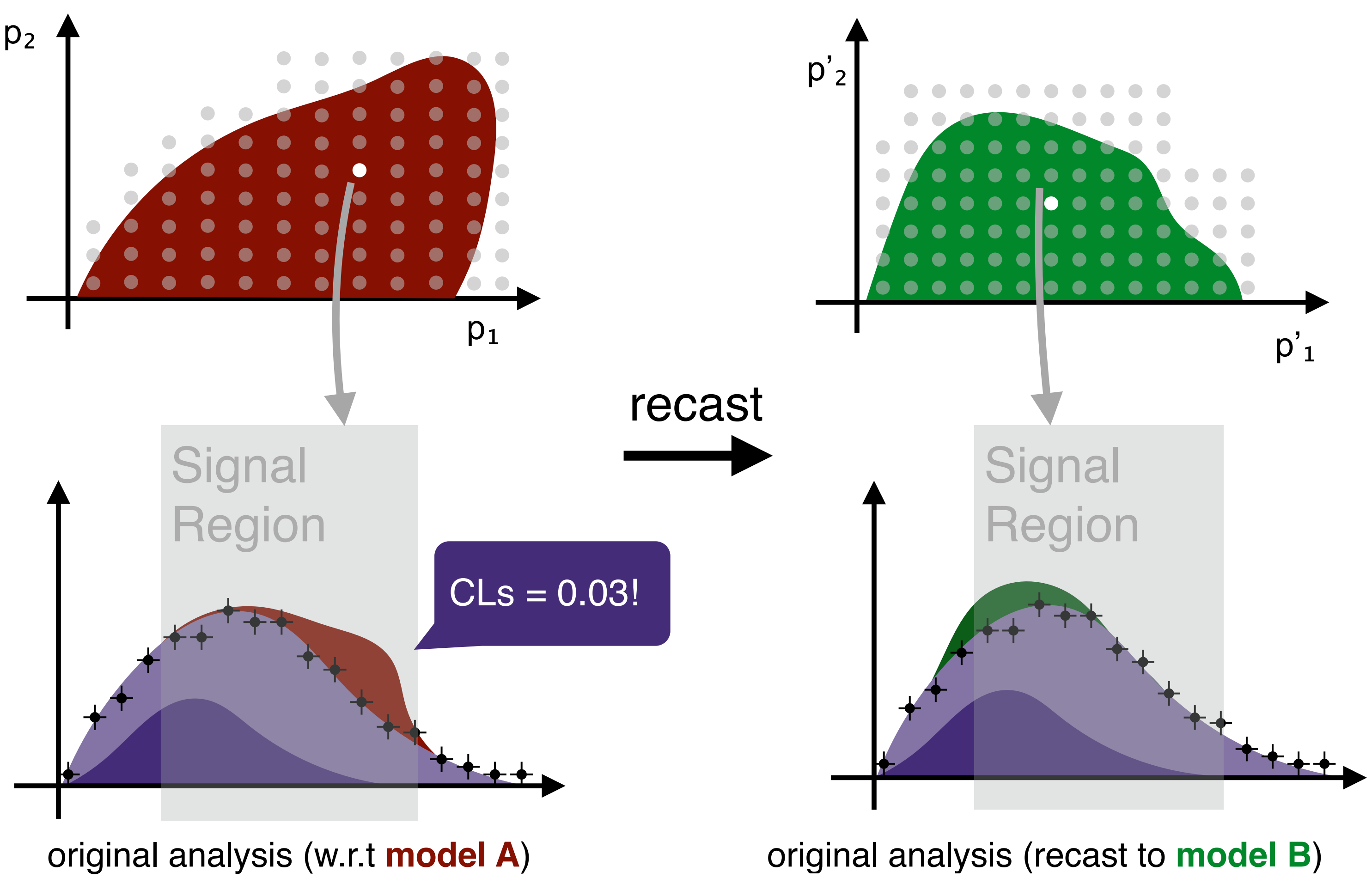
there are still unexcluded natural SUSY models!

the reinterpretation eco-system

There is a clear need for reinterpretation/recast of analyses to models beyond those matching to simplified models.

Three ingredients:

1. **Ability to generate new signal model**
2. **Access to implementation of event selection (incl. detsim, reco)[1]**
3. **Access to data and background distributions (incl. systematic variations)**



[1] unless analysis unfolded — but then other issues

the reinterpretation eco-system

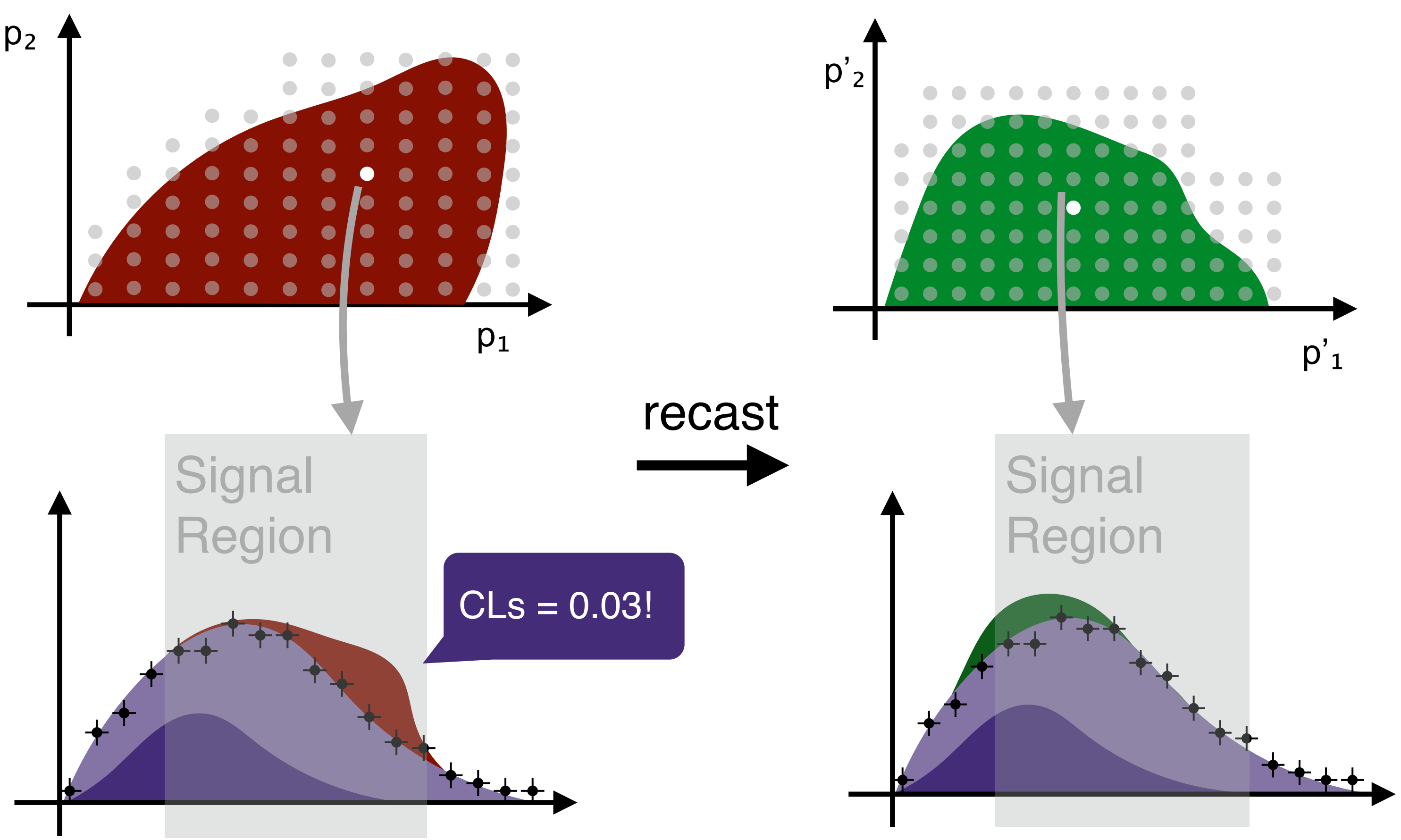
There is a clear need for reinterpretation/recast of analyses to models beyond those matching to simplified models.

can be done by theorists

Three ingredients:

1. **Ability to generate new signal model**
2. **Access to implementation of event selection (incl. detsim, reco)[1]**
3. **Access to data and background distributions (incl. systematic variations)**

can only *really* be done by experiments



original analysis (w.r.t **model A**)

original analysis (recast to **model B**)

[1] unless analysis unfolded — but then other issues

the reinterpretation eco-system

There is a clear need for reinterpretation/recast of analyses to models beyond those matching to simplified models.

Three ingredients:

1. **Ability to generate new signal model**
2. **Access to implementation of event selection (incl. detsim, reco)[1]**
3. **Access to data and background distributions (incl. systematic variations)**

To get around 2. and 3. theorists have developed a whole suite of tools.

- approximate detector simulation + reco (Delphes)
- approximate reimplementations of event selection using non-expt software stack (CheckMate, Rivet, …).
- approximate likelihoods from available background + data distributions, but mostly ignores systematics (e.g. HepData)
- try to get good results by getting experiments to release more data (efficiency maps, resolution parameters, acceptance tables, cutflows, etc…)

works very well for rough survey
but always only approximation
not on same footing as original result

ecosystem developed because of a lack of easy way to let experiments do it for theorists.

[1] unless analysis unfolded — but then other issues

the reinterpretation eco-system

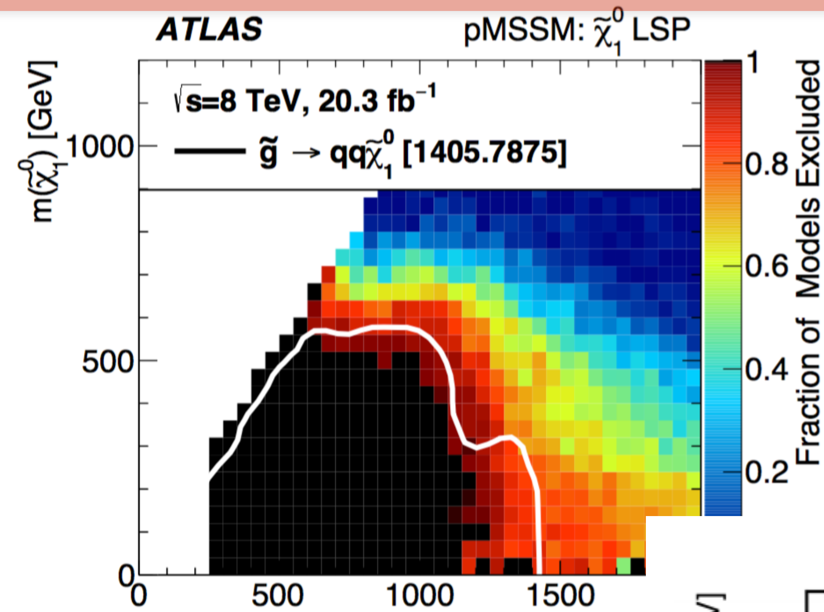
Historically it was **very hard** experiments to run a reinterpretation — even though theoretically it would have been possible. We do them, but rarely.

Three ingredients:

can only *really* be done by experiments

1. **Ability to generate new signal model**
2. **Access to implementation of event selection (incl. detsim, reco)[1]**
3. **Access to data and background distributions (incl. systematic variations)**

yes, but it's *hard*



JHEP

PUBLISHED FOR SISSA BY SPRINGER

RECEIVED: *August 27, 2015*
ACCEPTED: *September 23, 2015*
PUBLISHED: *October 21, 2015*

arXiv:1508.06608

19-D(!) pMSSM reinterpretation

Summary of the ATLAS experiment's sensitivity to supersymmetry after LHC Run 1 — interpreted in the

A re-interpretation of $\sqrt{s}$ = 8 TeV ATLAS results on electroweak supersymmetry production to explore general gauge mediated models

The ATLAS Collaboration

3-D recast for General Gauge Mediated SUSY Models

ATLAS-CONF-2016-033

JHEP

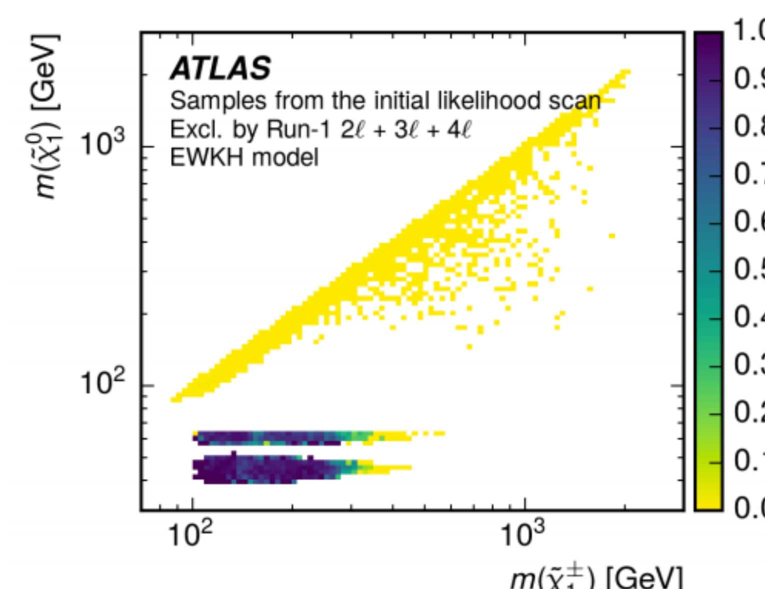
PUBLISHED FOR SISSA BY SPRINGER

RECEIVED: *August 3, 2016*
ACCEPTED: *September 22, 2016*
PUBLISHED: *September 30, 2016*

Dark matter interpretations of ATLAS searches for the electroweak production of supersymmetric parti

5-D scan of EWKH sector with help from STA

arXiv:1608.00872



the reinterpretation eco-system

Historically it was **very hard** experiments to run a reinterpretation — even though theoretically it would have been possible

Three ingredients:

1. **Ability to generate new signal model**
2. **Access to implementation of event selection (incl. detsim, reco)[1]**
3. **Access to data and background distributions (incl. systematic variations)**

can only *really* be done by experiments

yes, but it's *hard*

So, what's the hold up?

- well-oiled machine to generate new signals only matter of computing resources, priorities, but not a show-stopper ✔

the reinterpretation eco-system

Historically it was **very hard** experiments to run a reinterpretation — even though theoretically it would have been possible
Three ingredients:

1. **Ability to generate new signal model**
2. **Access to implementation of event selection (incl. detsim, reco)[1]**
3. **Access to data and background distributions (incl. systematic variations)**

can only *really* be done by experiments

yes, but it's *hard*

So, what's the hold up?

- well-oiled machine to generate new signals only matter of computing resources, priorities, but not a show-stopper ✔
- storage needs for data and background distributions is negligible. don't need full samples — just final distributions ✔

the reinterpretation eco-system

Historically it was **very hard** experiments to run a reinterpretation — even though theoretically it would have been possible
Three ingredients:

1. **Ability to generate new signal model**
2. **Access to implementation of event selection (incl. detsim, reco)[1]**
3. **Access to data and background distributions (incl. systematic variations)**

can only *really* be done by experiments

yes, but it's *hard*

So, what's the hold up?

- well-oiled machine to generate new signals only matter of computing resources, priorities, but not a show-stopper ✔
- storage needs for data and background distributions is negligible. don't need full samples — just final distributions ✔
- **up to know it was hard to preserve the bulk of the analysis beyond centralized signal generation. Solving analysis preservation enables new physics via reinterpretation**

Analysis Preservation in ATLAS

How hard can it be? Challenges for analysis preservation

- real ATLAS analyses are complex. Not a single file in a common framework (like e.g. Rivet, CheckMate, LHADA). **There's a reason have our own computing model.**
  - code is very diverse. many frameworks, scripts, etc..
- distributed teams, code, data: **one person rarely is able to run the entire analysis** pipeline — some develop event selection, some background estimates, some statistical analysis

To preserve analyses, we needed to respect the tools, workflows people use. instead of forcing a re-implementation, develop toolchain to capture what they are already doing.

1. capture software (*including all dependencies)* needed to run individual parts of an analysis (e.g. event selection) in a future-proof way.
2. capture logic how the many pieces of the analysis fit into an *analysis workflow* that can be re-executed on a new signal

Analysis Preservation in ATLAS

comprehensive software capture was intractable until recently (VMs??). Now progress in IT industry has **made it feasible — Linux Containers.** Technology with wide industry support — will be here for foreseeable future.

revolutionized software distribution & archival — "app store for generic software". Many additional tools that help deploy / run Linux Containers in "the cloud" (Google, Amazon, Microsoft, etc…).

Containers are now becoming a major topic in LHC collaborations. Simplifies a lot of our computing in many ways.

technology stack enabling realistic analysis preservation has become available recently

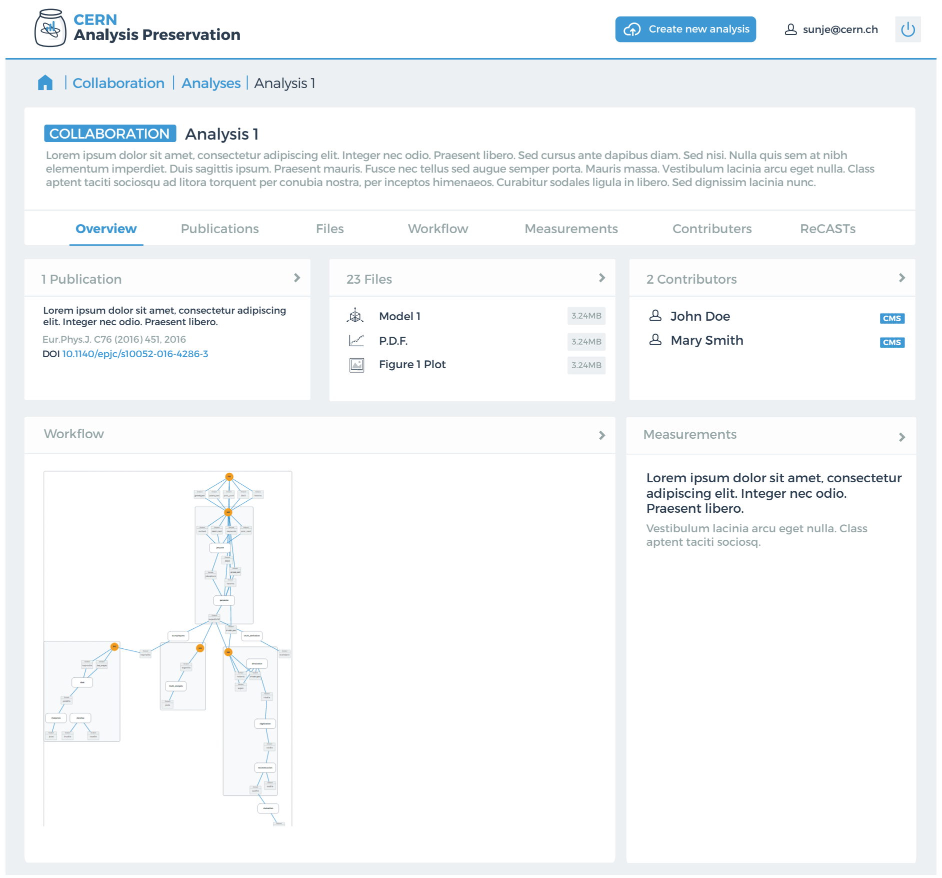
Analysis Preservation in ATLAS

CERN is committed to analysis preservation. CERN Analysis Preservation (CAP) portal being built by CERN IT and Information Services devisions.

ATLAS-developed workflow language, natively integrated. Built to allow anyone in the collaboration to re-run a specific analysis using information stored inside of CAP.

workflow	software	data
eventsel.yml	docker img	data, bkgds
fit.yml	docker img	
workflow.yml		

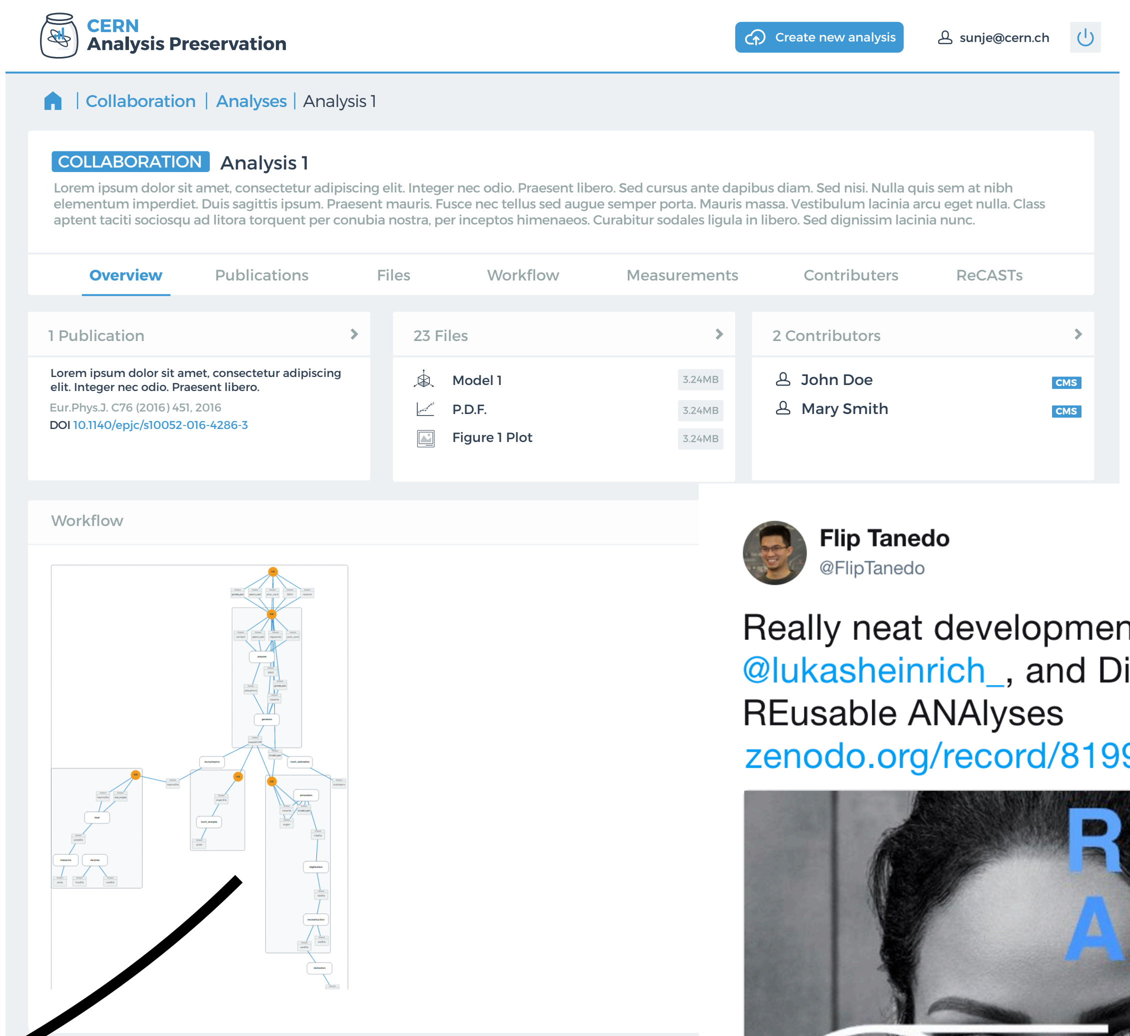
import analysis workflow



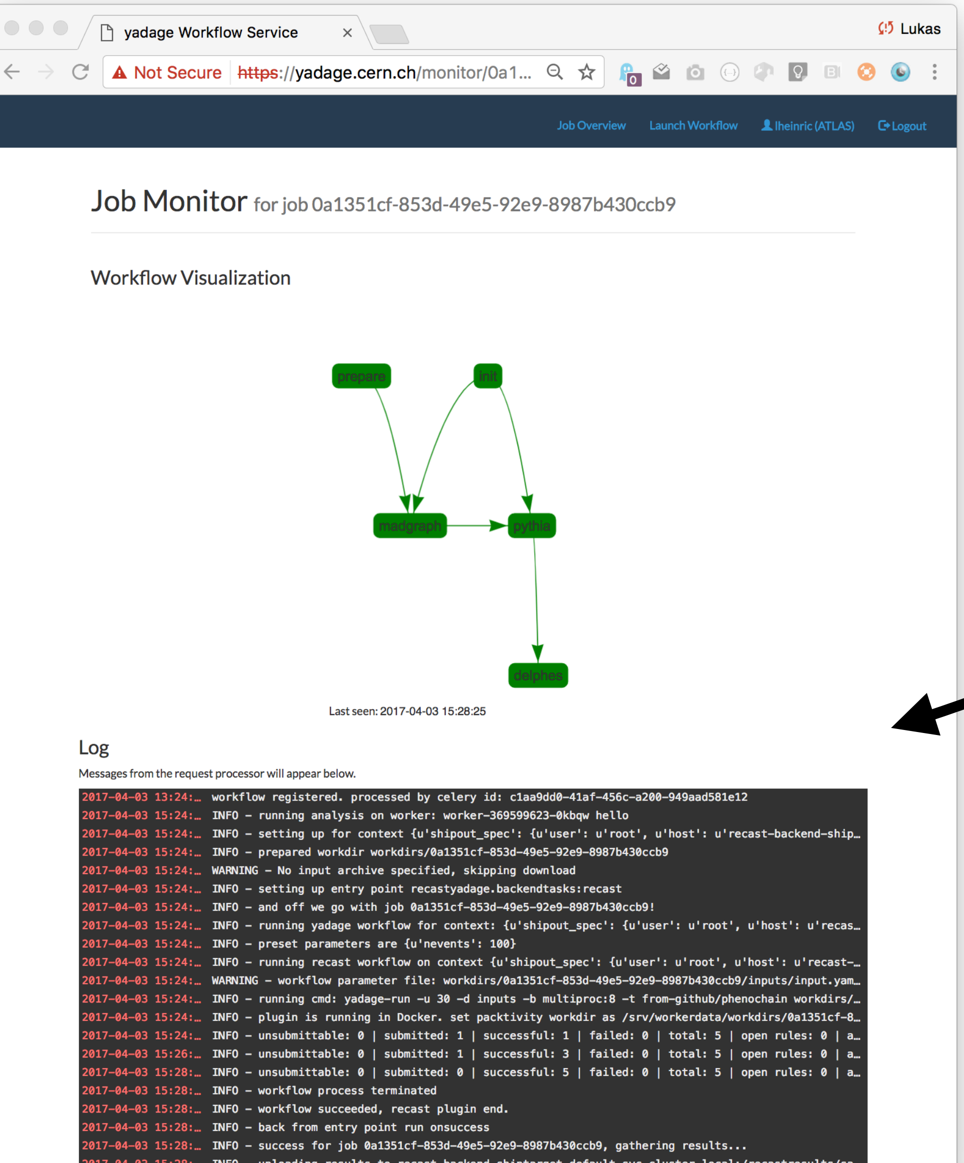
CERN Analysis Preservation

Analysis Preservation in ATLAS

major pillar in CAP: cloud-based infrastructure/service **REANA** to re-use/re-execute anlyses stored in CAP at scale.



correction: many more people involved!



extract analysis information (code, data) and re-execute on new input on a cluster

CERN Analysis Preservation

Analysis Preservation is within reach.

Towards a streamlined RECAST service

RECAST

High Energy Physics has always been leading when it comes to internet-enabled collaboration and services: arXiv, SPIRES, INSPIRE, HepData,

With archived analysis workflows it becomes feasible to streamline the reinterpretation efforts.

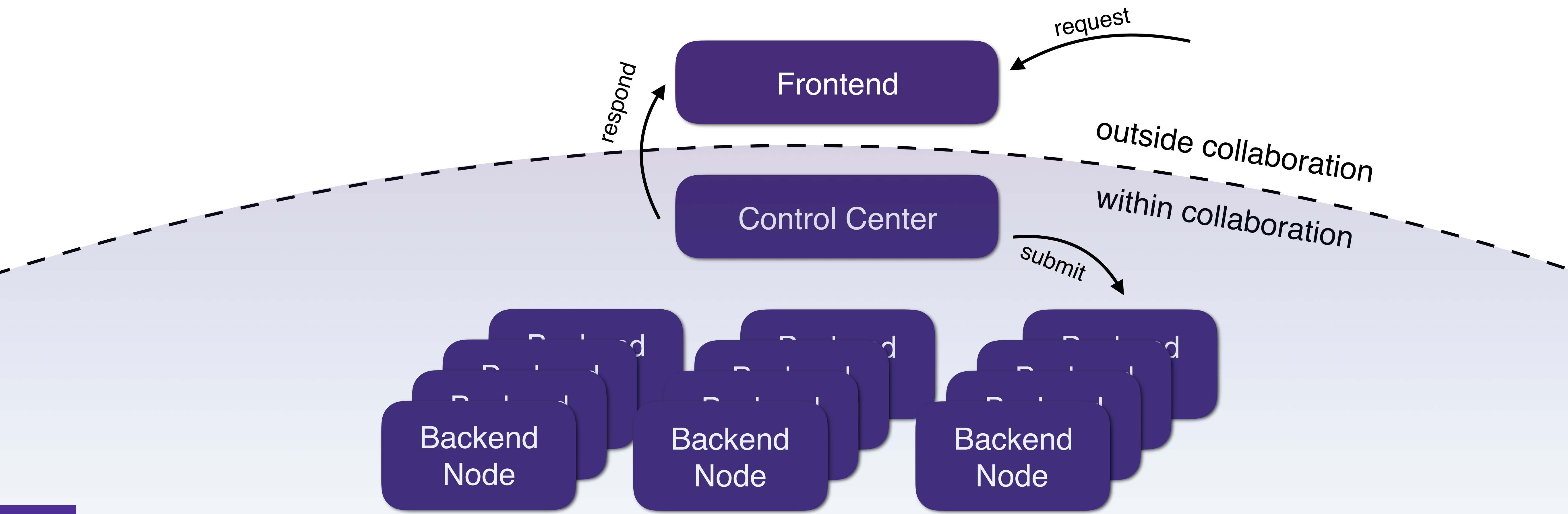
Reinterpretations as a community-wide service.

- **Enables interaction with LHC data for people outside of collaboration without experiments releasing the data.**
- **Produces authoritative results backed by the collaboration.**

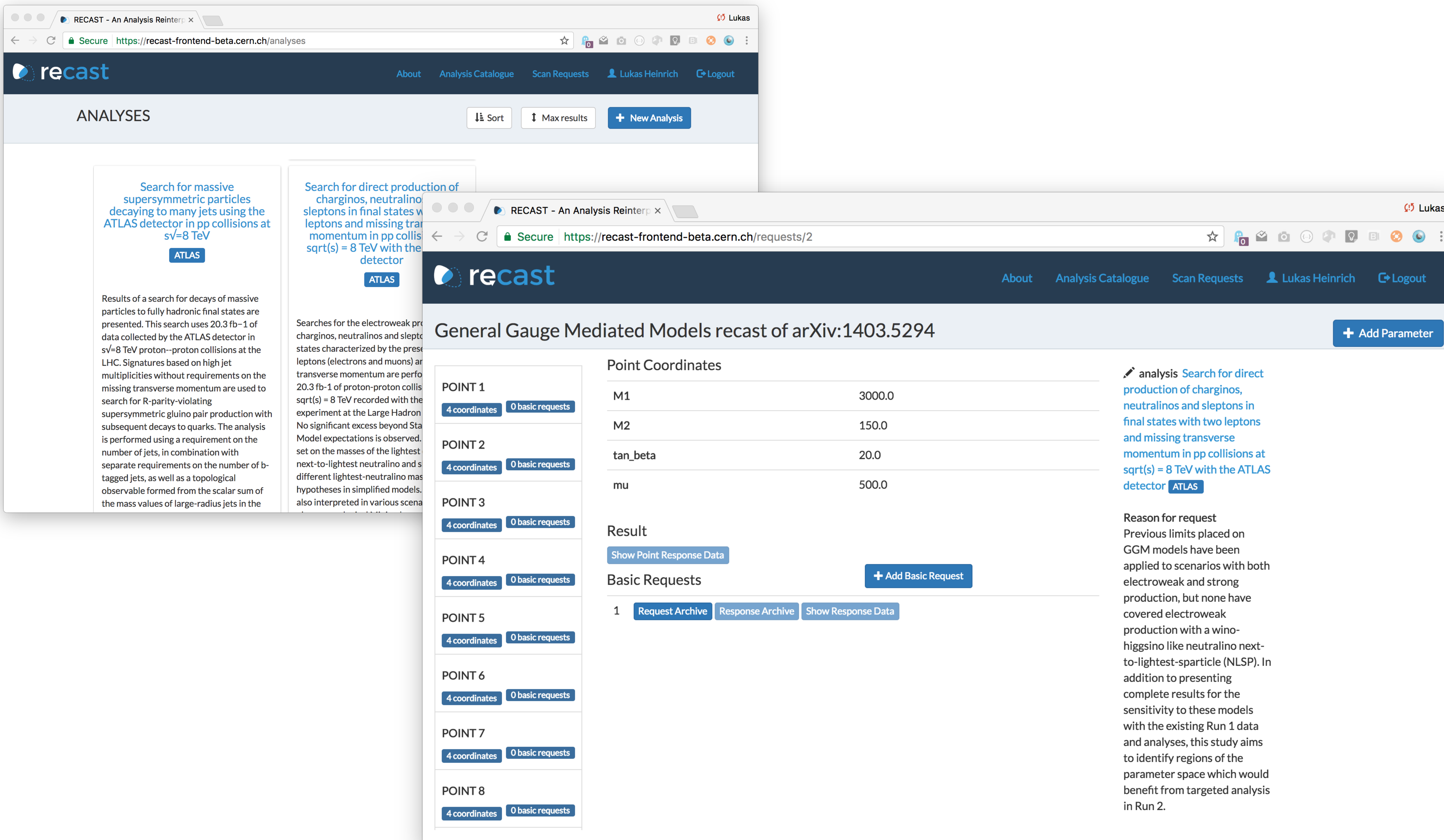
2010! it's been a long time…

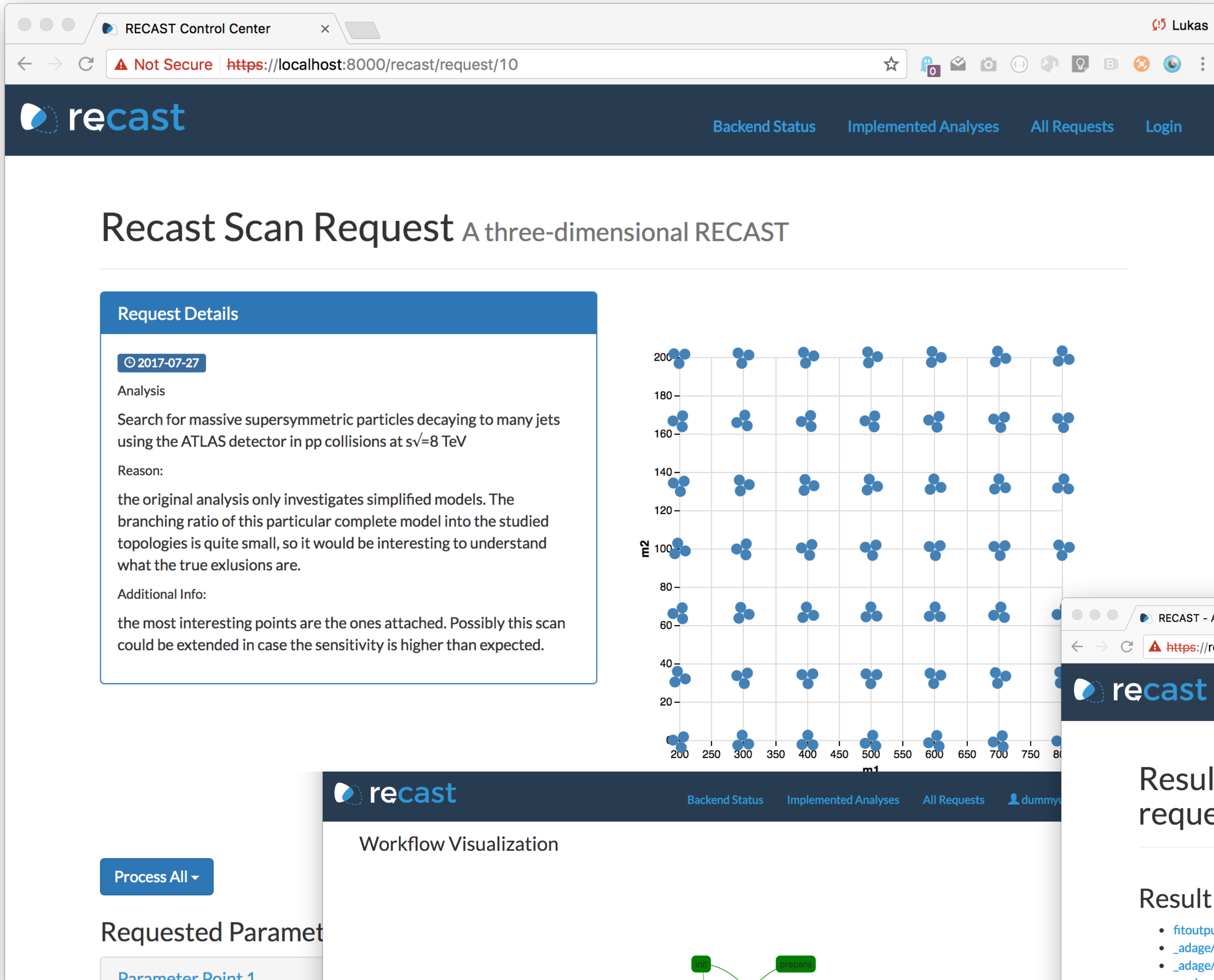
Cranmer, Yavin [arXiv:1010.2506])

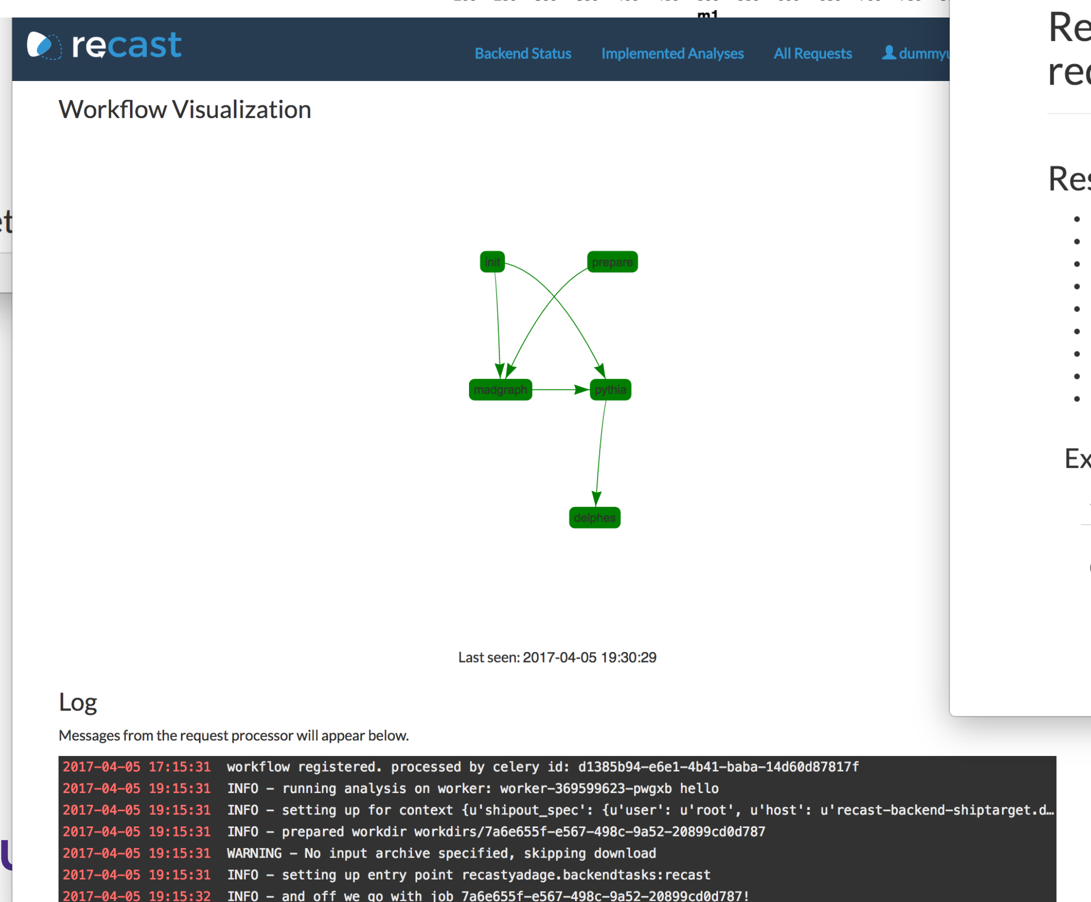
- Produce reinterpretations of same fidelity as original result (not just approximations)
- Allow hep-ph community to *suggest* reinterpretations through a standard (web) interface. They provide most interesting points / scans to do. Auxiliary information such as run cards, SLHA spectra, UFO models
- LHC collaborations review suggestions and choose which to fulfill (based on scale of request, availability of a preserved analysis, physics case)
- Use archived analysis to (semi-) automatically run reinterpretation. Review results, approve (possibly on accelerated track, since analysis already approved).
- Publish and/or append original analysis HEPDATA record.
- Allows us to decouple original publication from reinterpretations. Publish early using benchmark signals, continuously re-interpret as samples become available

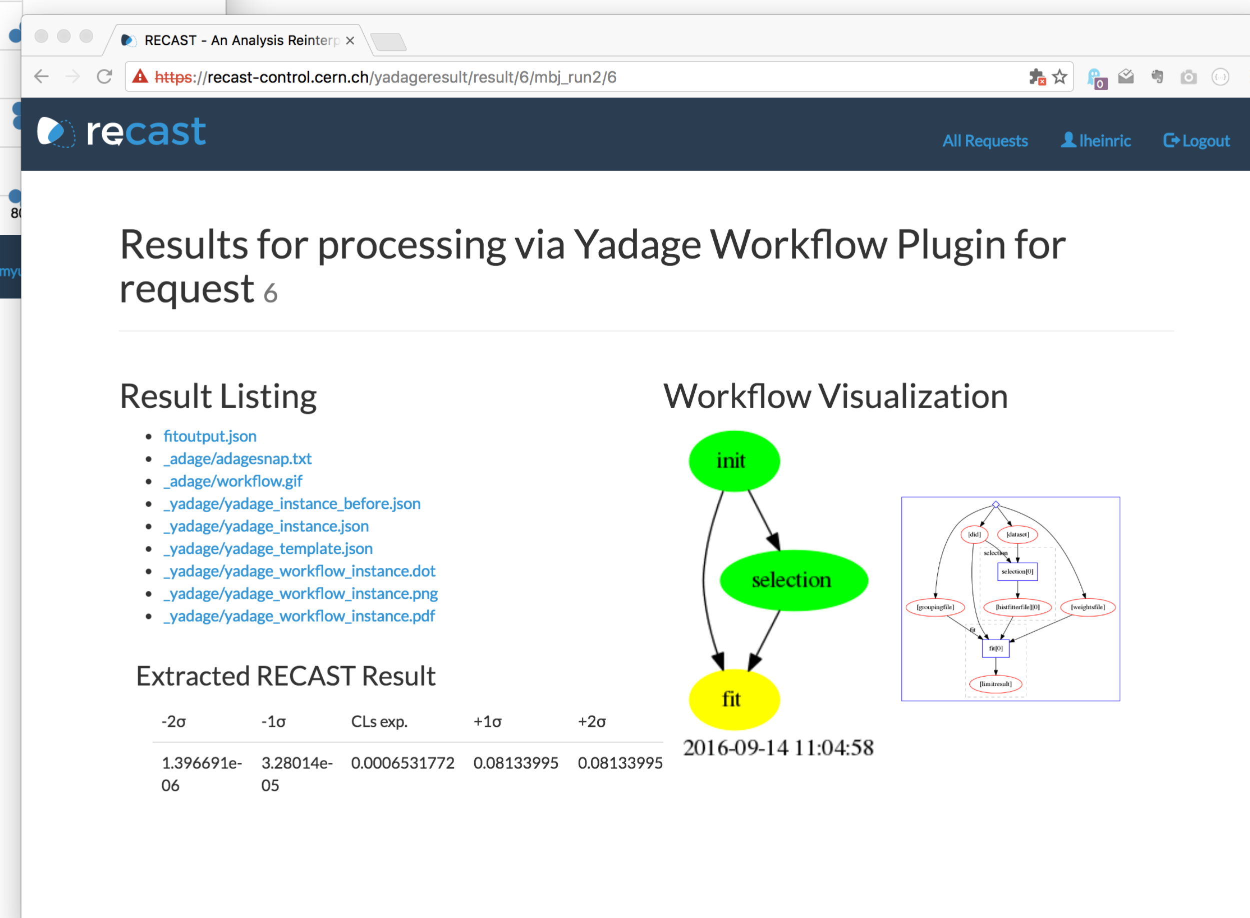


public-facing RECAST service to let theorists suggest new models, upload necessary data (parameter/SLHA files, UFO models, etc..)









collaboration internal dashboard

- to view requests
- execute analysis workflows
- inspect results
- upload response

RECAST Outlook

we’re still learning — but results look very promising. cloud infrastructure has been used for a number of reinterpretations (pMSSM, DM recast)

now trying to mainstream use of analysis preservation within the collaboration. Prepare for full-dataset analyses, summary papers that based on reinterpretation (a la pMSSM scan)

Recent progress in IT technology makes full-fidelity analysis preservation finally feasible.

Reinterpretation is killer app of Analysis Preservation — new science through reusable analyses (reproducibility of original results comes for free)

ATLAS, CERN building infrastructure to leverage technology and enable cloud-based reinterpretation: CAP, REANA

If internal use proves successful, good chance to offer RECAST as a reinterpretation service.


ATLAS analysis
ATLAS analyses
checking
one model
It's the difference between if you had airplanes
where you threw away an airplane after every flight,
versus you could reuse them multiple times.
— Elon Musk

Tips / Remarks

- When code is in source control (git/svn), with clear installation instructions, usually capture is not painful
- Many analyses are moving code to GitLab. GitLab has continuous integration built-in. If this is used but he analysis team, eases process considerably
- Usually fitting code needs most adjustments.
  - Models/Grids are hardcoded (e.g. assumptions on names like “myModel_m123_m938”).
  - needs to be able to run an single arbitrary model.
- **Typical amount of work per analysis: ~ 1 week of coordinated work with analysis experts to capture analysis. Expect to become easier as we collect experience.**

Goals

Feedback to task force by OAB: more examples, please!

My suggestion:

- Attempt to capture a handful of "pilot analyses" in the Exotics working group leading up to summer
- Identify possible existing BSM datasets that are known to have acceptance in captured analyses
- if applicable, generate new signal samples, in light of archived analyses (with input from theorists)
- already plans within **LLP** sub-group to identify suitable analysis. **Mono-H** good candidate, other suggestions?

Input from Exotics WG important

- Exotics analysis workflows different from typical SUSY analysis
- Exotics analyses often rely on non-traditional reconstruction object. Hard to provide e.g. simple efficiency maps, etc. To reinterpret correctly need FullSim signal samples + original analysis workflow. Perfect use-case for RECAST

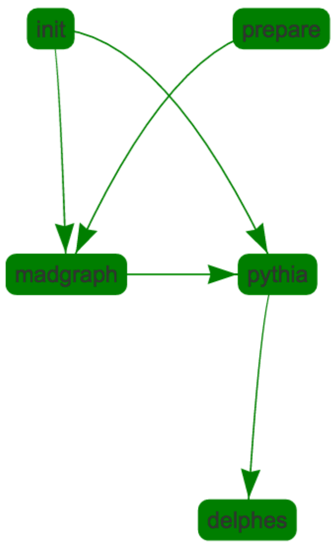
Happy to organize 1-/2-day workshop/hackathon to work with multiple analyses in unison.

Appendix

recast

Backend Status | Implemented Analyses | All Requests | dummyuser | Logout

Workflow Visualization



Last seen: 2017-04-05 19:30:29

Log

Messages from the request processor will appear below.

```
2017-04-05 17:15:31  workflow registered. processed by celery id: d1385b94-e6e1-4b41-baba-14d60d87817f
2017-04-05 19:15:31  INFO - running analysis on worker: worker-369599623-pwgxb hello
2017-04-05 19:15:31  INFO - setting up for context {u'shipout_spec': {u'user': u'root', u'host': u'recast-backend-shiptarget.d…
2017-04-05 19:15:31  INFO - prepared workdir workdirs/7a6e655f-e567-498c-9a52-20899cd0d787
2017-04-05 19:15:31  WARNING - No input archive specified, skipping download
2017-04-05 19:15:31  INFO - setting up entry point recastyadage.backendtasks:recast
2017-04-05 19:15:32  INFO - and off we go with job 7a6e655f-e567-498c-9a52-20899cd0d787!
2017-04-05 19:15:32  INFO - running yadage workflow for context: {u'shipout_spec': {u'user': u'root', u'host': u'recast-backen
```

Analysis as a function mapping data and models to results

$$\text{result} = f_{\text{analysis}}(\text{data}|\text{model})$$

observable distributions, confidence intervals on model parameters

reconstruction, event selection, stat. evaluation

collision data from LHC detector

model hypothesis (SM, many SUSY models, etc..)

archive analysis in a **parametrized form**, such that we can quickly run a new model

$$f_{\mathrm{analysis}}(\mathrm{data}|\,\cdot\,)$$

Given a **parametrized preservation of an analysis** (even w/ fixed data), we gain ability to extract **new results** using existing resources.

Reinterpretation of Single Analysis under multiple models

Combination of multiple analyses w.r.t. one model
(increased stat. power)

$$f_{\mathrm{a}}(\mathrm{data}|\mathrm{model}_1) \quad f_{\mathrm{a}}(\mathrm{data}|\mathrm{model}_2)$$

$$f_{\mathrm{b}}(\mathrm{data}|\mathrm{model}_1) \quad f_{\mathrm{b}}(\mathrm{data}|\mathrm{model}_2)$$

$$f_{\mathrm{c}}(\mathrm{data}|\mathrm{model}_1) \quad f_{\mathrm{c}}(\mathrm{data}|\mathrm{model}_2)$$

Case Study: Multi B-jets analysis

Defining the individual Workflow steps

- need script that tell us how to run the code once we are in the right environment. parametrized by a few variables (input file names etc)
- can use simple shell script, but also anything else

lumi/xsec/KF/FE weighting of HF tree

23 lines (22 sloc) | 714 Bytes — Raw | Blame | History

```
process:
  process_type: 'interpolated-script-cmd'
  script: |
    #!/bin/bash
    echo "Hello"
    source ~/.bashrc
    setupATLAS
    source ./rcSetup.sh
    /recast_auth/getmyproxy.sh
    lsetup fax dq2
    python MultibjetsAnalysis/scripts/Run.py --dataSource 1 --doSyst 1 --doNTUPSyst 1 --doNTUP 0 --doxAOD 0 --doH
    mv {submitdir}/data-output_histfitter/*.root {outputprefix}.{did}.root
publisher:
  publisher_type: 'fromglob-pub'
  globexpression: '*.root'
  outputkey: histfitterfile
environment:
  environment_type: 'docker-encapsulated'
  image: lukasheinrich/multibsel_cvmfs
  resources:
    - CVMFS
    - GRIDProxy
```

49 lines (42 sloc) | 1.42 KB — Raw | Blame | History

```
process:
  process_type: 'interpolated-script-cmd'
  script: |
    #!/bin/bash
    source ~/.bashrc
    setupATLAS
    lsetup "root 6.06.02-x86_64-slc6-gcc48-opt"
    cd /code/multib/HistFitter
    source ./setup.sh
    cd analysis/analysis_multib


    /recast_auth/getkrb.sh
    #klist
    #exit
    cd input
    python mergeTrees.py {selectionoutput} --filters filters/filters_ht.json --weights {weightsfile} --did-to-group {groupingfi
    cd ..

    lumi="5807.51"
    grid="{gridname}"
    region="Gbb_A"
    tag="tag2.4.11-1-0_July00"
    echo '["{pointname}"]' > point.json
    cat point.json
    export HF_MBJ_SIGNALJSON="point.json"
    export HF_MBJ_BACKGROUNDFILE={bkgtree}
    export HF_MBJ_DATAFILE={datatree}
    export HF_MBJ_SIGNALFILE='input/Sig.root'
    HistFitter.py -wtpf -F excl python/My3bGtt.py _signalRegion $region _lumi $lumi _unblind true _doHFSplitting false 2>&1 | t

    resultfile=$(ls results/My3bGtt_*fixSigXSecNominal*_hypotest.root)
    echo "result file is:  $resultfile"
    root -b -q 'root2json.C("'"$resultfile"'","hypo_Gbb_%f_%f")'

    jsonfile=$(ls *harvest_list.json)
    python recast_format.py $jsonfile {outputjson}
```

direct SH Driver reads signal dataset (a SUSY10 derivation) via XrootD writes out HistFitter tree

Run HF

Extract Results into JSON format

Case Study: Multi B-jets analysis

Stringing the workflow together

- small file on how the individual pieces fit together.
- Here: dataset, AMI info file etc provided as input parameters, define EOS location of signal and background trees, declare that signal histfitter tree comes from previous selection step etc

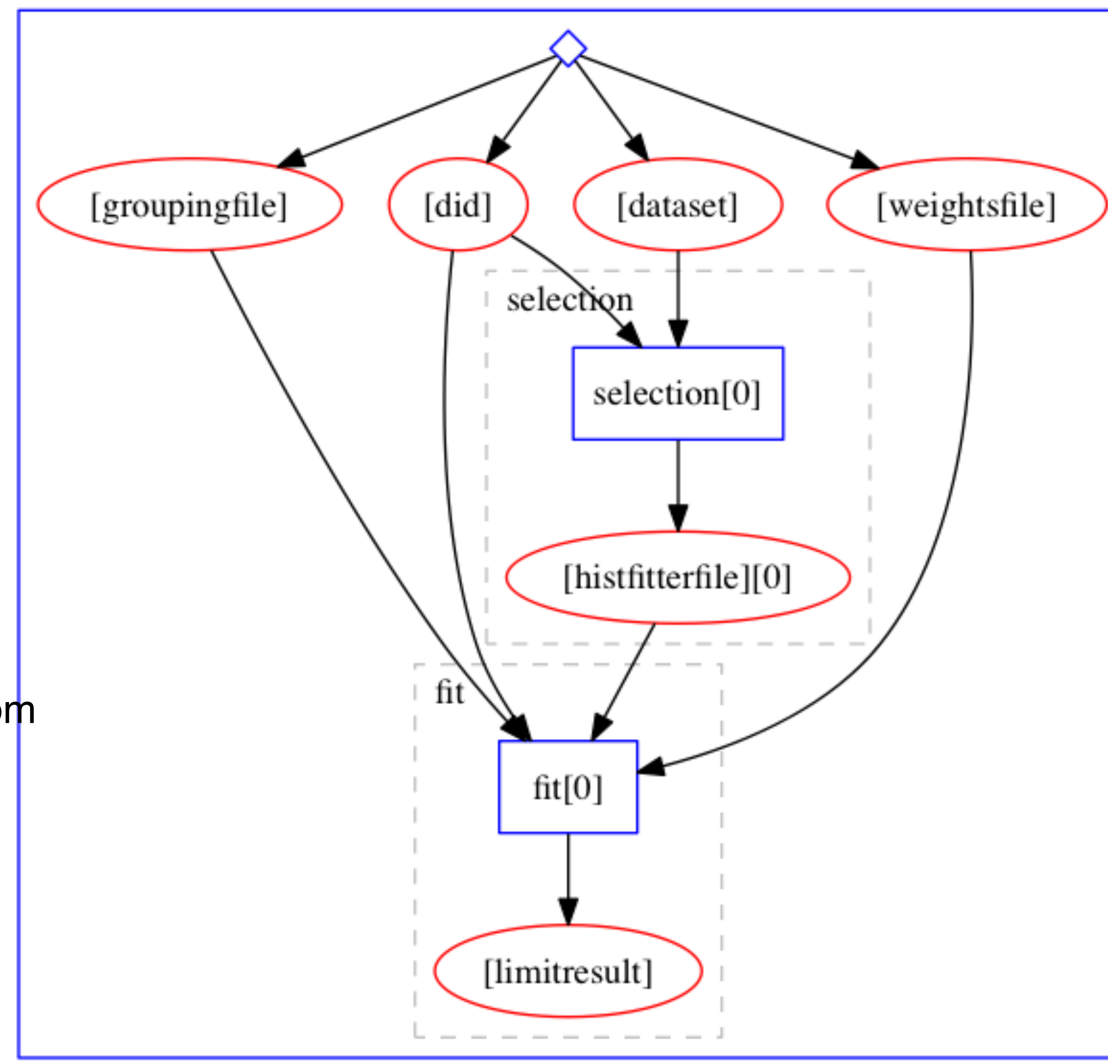
27 lines (26 sloc) | 1.07 KB

Raw | Blame | History

```yaml
stages:
  - name: selection
    dependencies: ['init']
    scheduler:
      scheduler_type: singlestep-stage
      parameters:
        dataset: {stages: init, output: dataset, unwrap: true}
        submitdir: '{workdir}/submitdir'
        outputprefix: '{workdir}/histfitter.root'
        did: {stages: init, output: did, unwrap: true}
      step: {$ref: 'selscript.yml#'}
  - name: fit
    dependencies: ['selection']
    scheduler:
      scheduler_type: singlestep-stage
      parameters:
        bkgtree: 'root://eosuser.cern.ch///eos/project/r/recast/Bkg_2.4.15-2-0_merged.root'
        datatree: 'root://eosuser.cern.ch///eos/project/r/recast/Data_2.4.15-2-0.root'
        outputjson: '{workdir}/fitoutput.json'
        pointname: 'Gbb_1600_200'
        gridname: Gbb
        selectionoutput: {stages: selection, output: histfitterfile, unwrap: true}
        weightsfile: {stages: init, output: weightsfile, unwrap: true}
        groupingfile: {stages: init, output: groupingfile, unwrap: true}
        did: {stages: init, output: did, unwrap: true}
      step: {$ref: 'fitscript.yml#'}
```

data and background trees archived in access-controlled location

take signal HF tree from previous step



How to preserve $f_{\text{analysis}}(\,\cdot\,)$?

1. Problem: Preserve Individual Processing Steps

(Example: Run Detector Simulation + Reconstruction on MC events)

Steps (“activities”) process data obtained by a global state, and modify state with (eg. writing new files, modify existing files)

$$\text{result data},\ \text{state}' = g_{\text{step}}(\text{state}, \text{parameters})$$

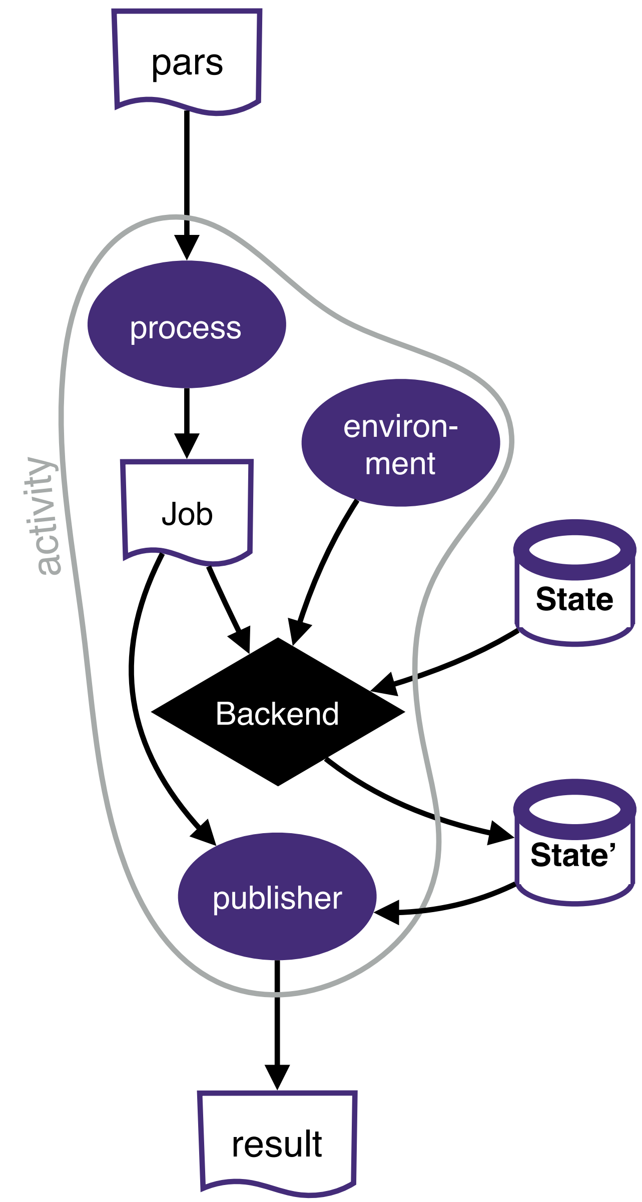
It’s useful to have machine readable result data to e.g. identify newly created files.

Three ~orthogonal ingredients that can be described individually:

parametrized process:
template job from which we can produce concrete job
template: “`./DelphesHepMC <input file> <output file>`”
concrete: “`./DelphesHepMC /input/file/path.hepmc /output/file.root`”

environment:
description of computing env in which above job can run. Multiple options, promising: *Linux Containers* (investigating Umbrella, etc)

publisher:
recipe how to extract parsable result data after job completion
e.g. globbing files in a work directory



How to preserve $f_{\text{analysis}}(\,\cdot\,)$?

1. Problem: Preserve Individual Processing Steps

(Example: Run Detector Simulation + Reconstruction on MC events)

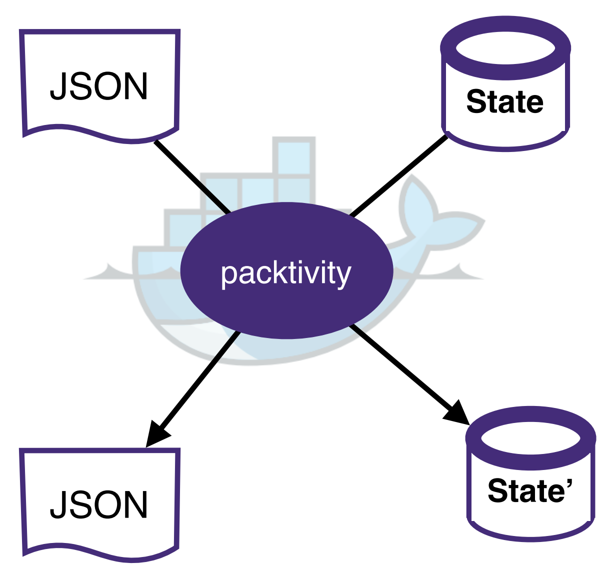
Data Format: JSON

- as interchange format for parameters and result data
- as declarative description format for *process/env/publisher*
  - incl. JSON schemas for validation

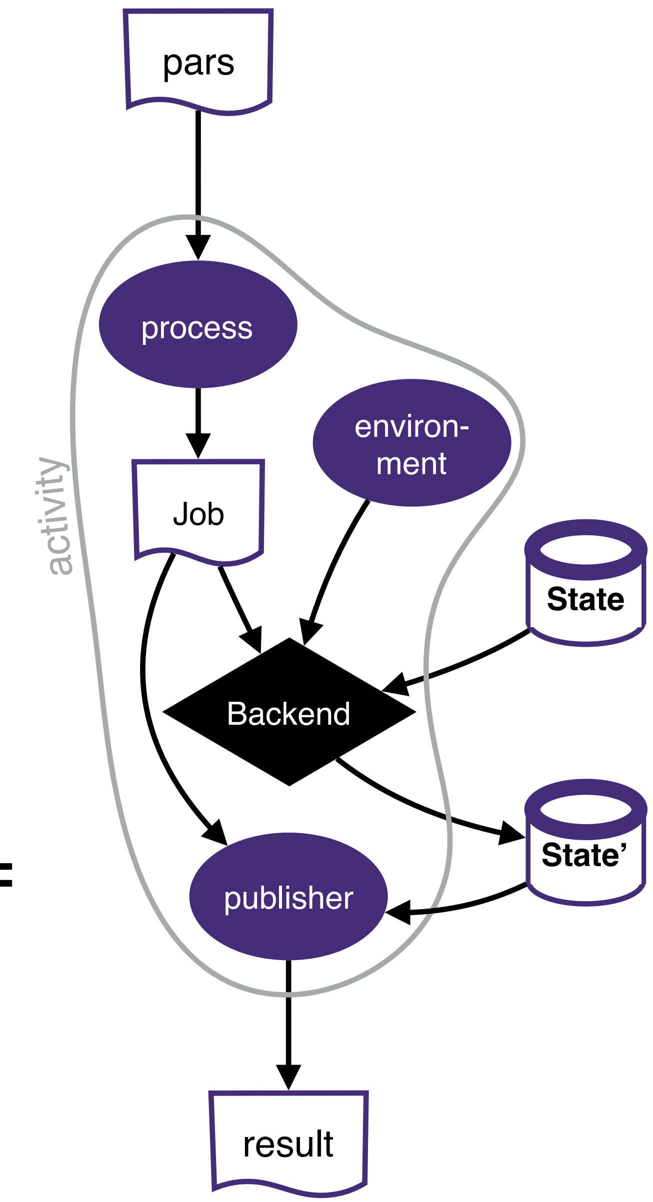
Essentially, a self-consistent “packaged activity” — a “packtivity"

- JSON API
- archivable, declarative description as JSON
- dependencies captured in environment
  - e.g. Docker Image

$$\text{result data},\ \text{state}' = g_{\text{step}}(\text{state}, \text{parameters})$$ =



=



How to preserve $f_{\text{analysis}}(\,\cdot\,)$?

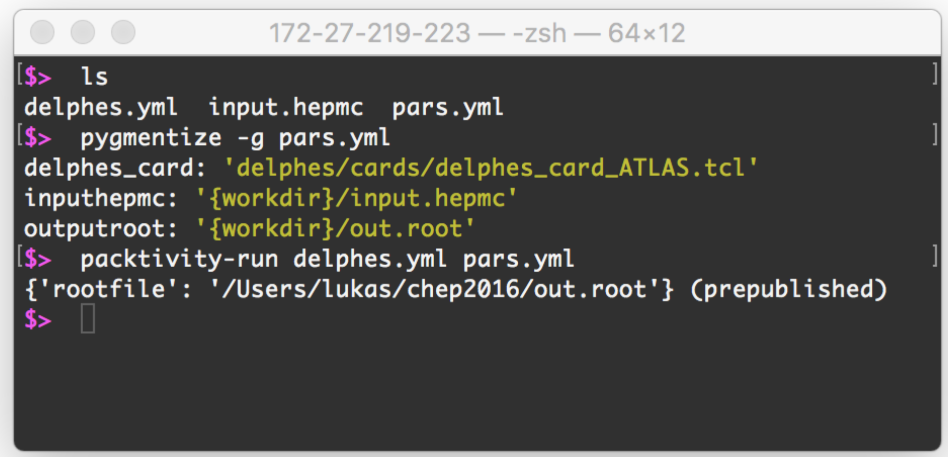
1. Problem: Preserve Individual Processing Steps

(Example: Run Detector Simulation + Reconstruction on MC events)

Example:

```
process:
  process_type: 'string-interpolated-cmd'
  cmd: 'DelphesHepMC  {delphes_card} {outputroot} {inputhepmc}'
publisher:
  publisher_type: 'frompar-pub'
  outputmap:
    rootfile: outputroot
environment:
  environment_type: 'docker-encapsulated'
  image: lukasheinrich/root-delphes
```

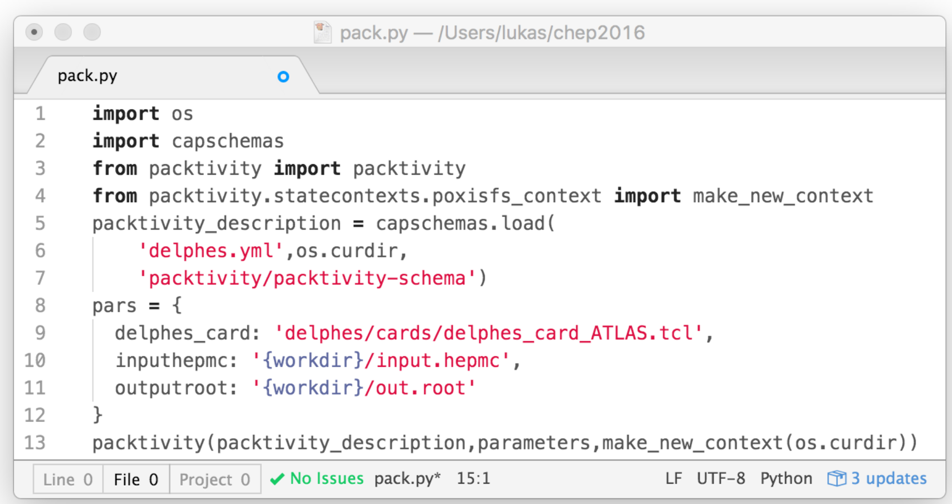
CLI tool

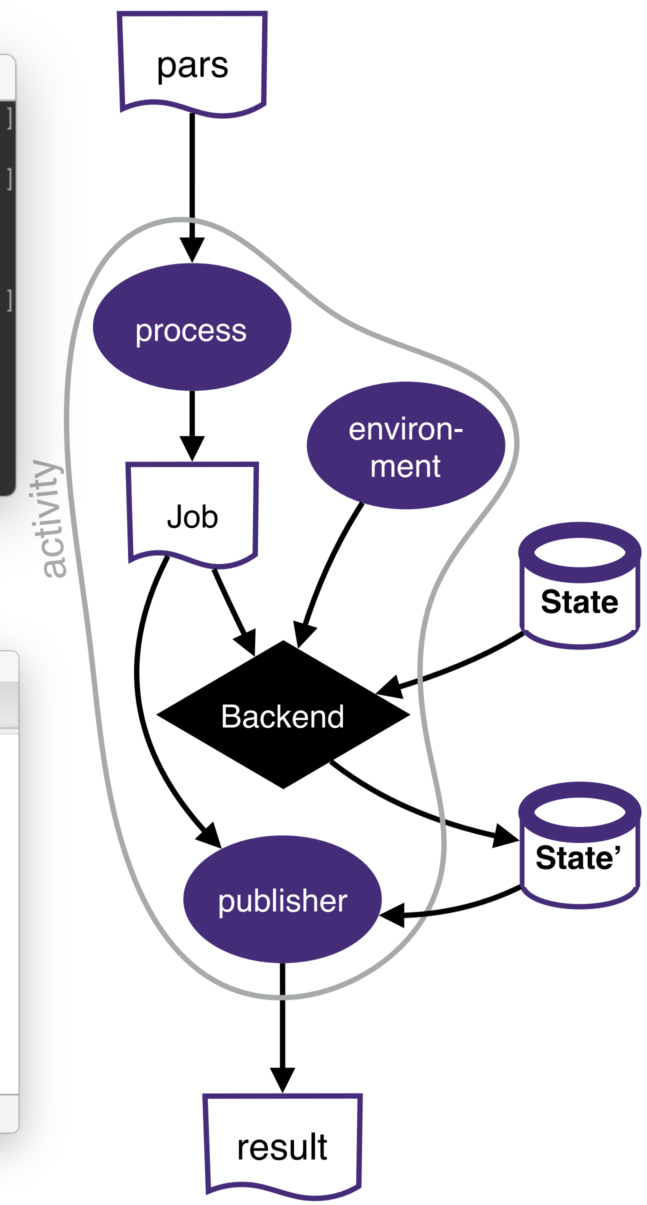


python package: “packtivity”

- executes packtivities according to JSON spec for given parameters
- cli tool and python bindings
- multi-host / remote execution ready via e.g. Docker Swarm

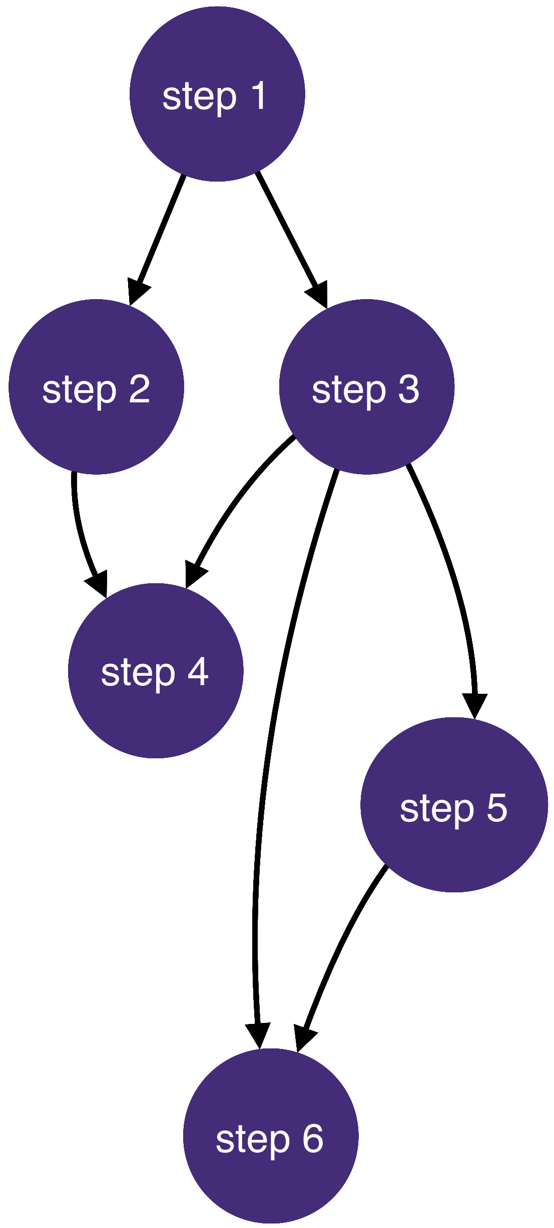
python bindings





How to preserve $f_{\text{analysis}}(\,\cdot\,)$?

2. Problem: Preserve Parametrized Workflow



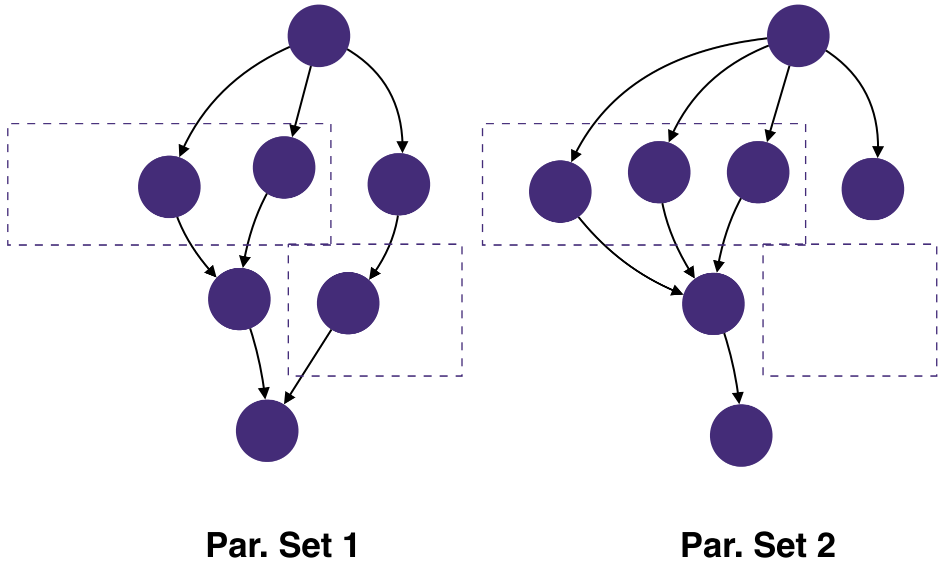
Natural Data Model: *directed acyclic graphs (DAGs)*

- **nodes**: individual steps
- **edges**: dependency relations

Two place where parametrization enter:

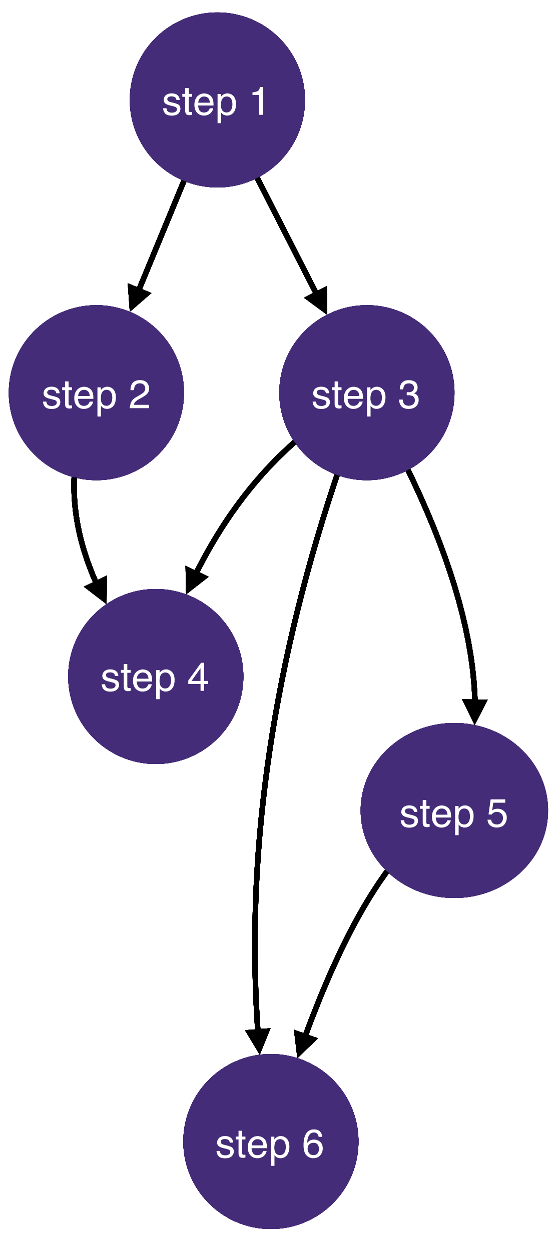
1. individual steps parametrized: covered by “packtivities”
   graph topology may *depend on the parameters* of the analysis and only emerge during run-time

2. Examples:
   - variable number of created files during execution,
   - conditional choices (if/else)/flags do enable/disable steps, e.g. run systematics / not



How to preserve $f_{\text{analysis}}(\cdot)$?

2. Problem: Preserve Parametrized Workflow



Therefore: Sequentially build up graph, as sufficient information becomes available, using a number of stages that add nodes and edges

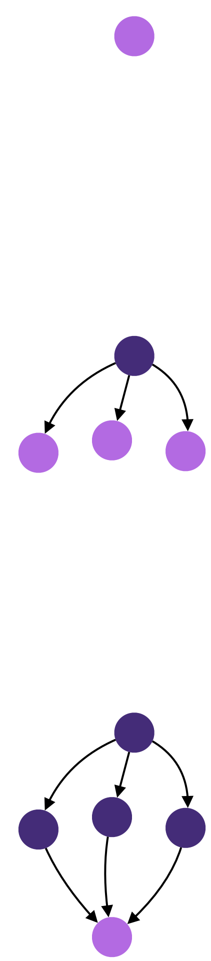
To capture analysis workflow, capture the stages.

Example: Parametrized Map-Reduce

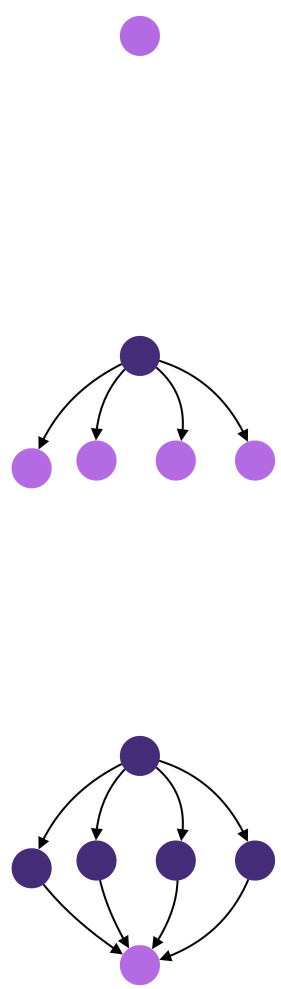
Stage 1:
unknown number of files. e.g. download & unpack archive with a priori unknown # of files

Stage 2:
for each file in the archive, add node to process it
(**only possible after first node done)**

Stage 3:
add a node that merges results of the map nodes
node/edge can be added before execution of map nodes



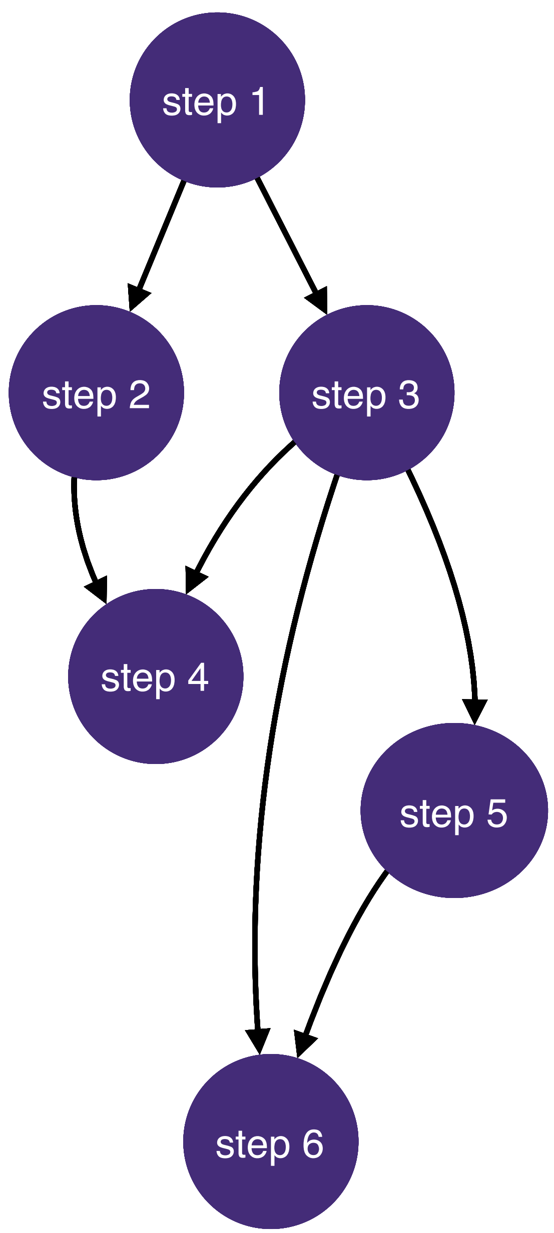
Par. Set 1



Par. Set 2

$$f_{\text{analysis}}(\,\cdot\,)$$

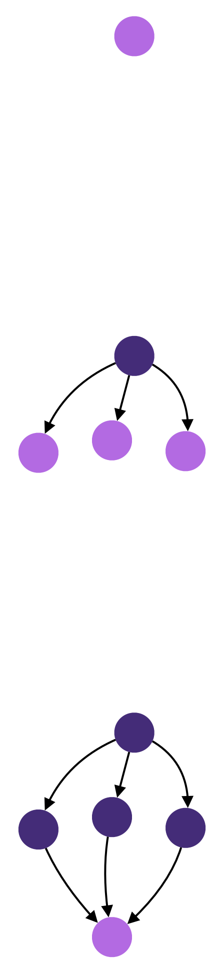
2. Problem: Preserve Parametrized Workflow



Therefore: Sequentially build up graph, as sufficient information becomes available, using a number of stages that add nodes and edges

To capture analysis workflow, capture the stages.

Example: Parametrized Map-Reduce

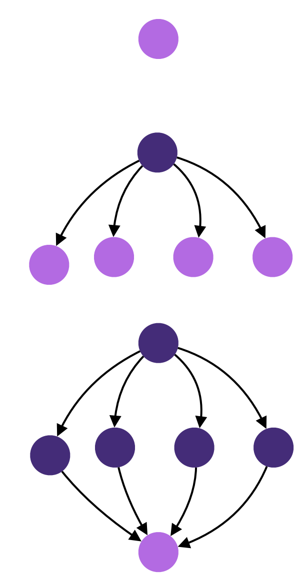


Par. Set 1

Stage 1:
unknown number of files. e.g. download & unpack archive with a priori unknown # of files

Stage 2:
for each file in the archive, add node to process it
(**only possible after first node done)**

Stage 3:
add a node that merges results of the map nodes
node/edge can be added before execution of map nodes



Par. Set 2