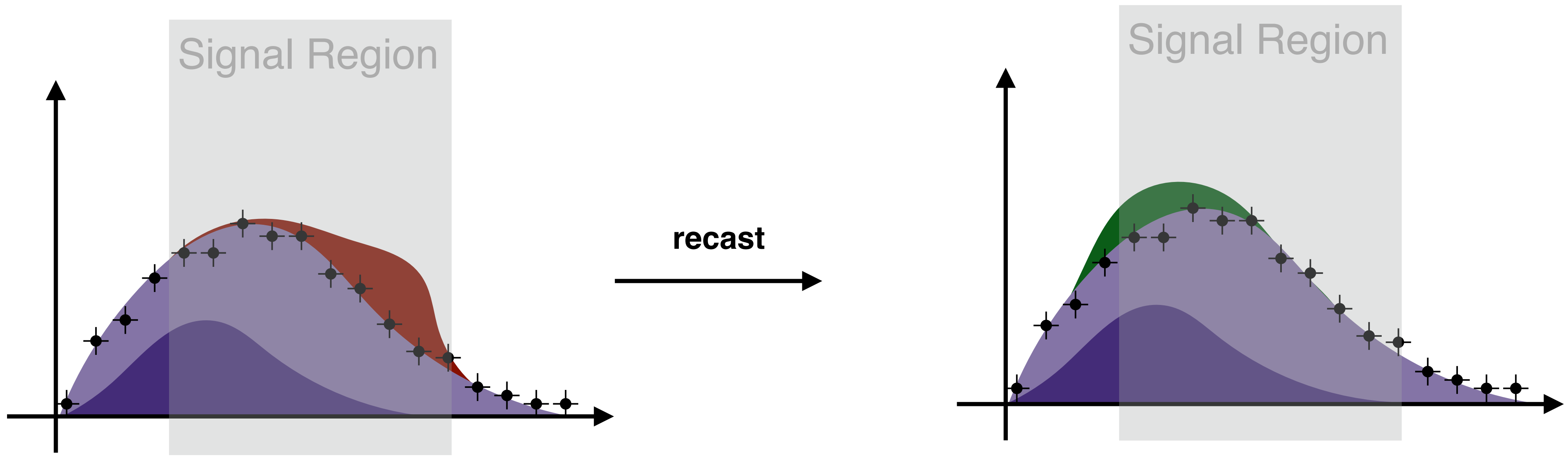
RECAST

Lukas Heinrich

LLP Prep Meeting

NEW YORK UNIVERSITY

Reinterpretation: interpreting an existing experimental result in the context of an alternative BSM physics scenario



original analysis (w.r.t **model A**) original analysis (recast to **model B**)

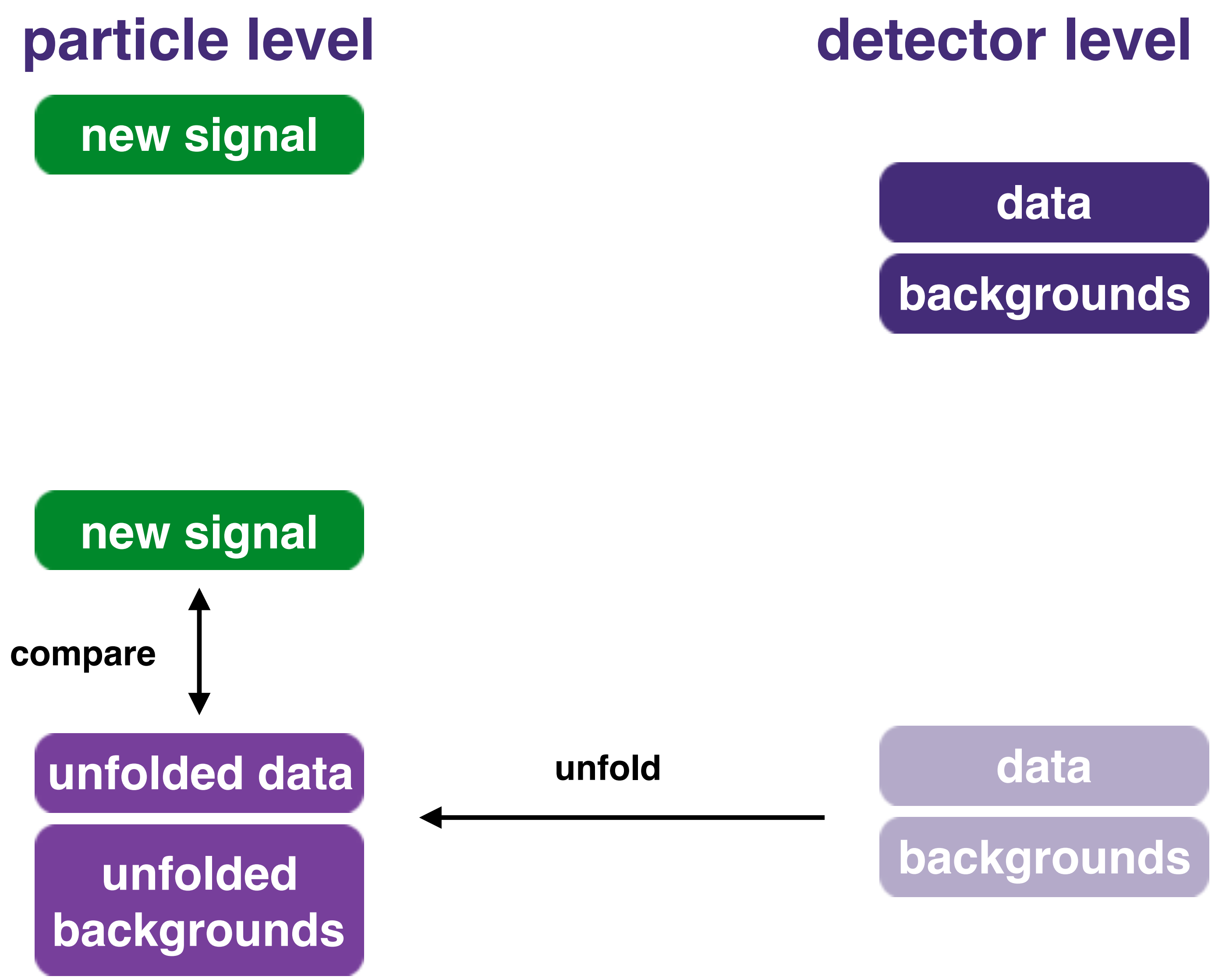
The **need for reinterpretation** is driven mainly by

- cost of creating a new, dedicated analysis for a new model vs re-use of old analysis
- relative scarcity of experimental results vs relative abundance of theoretical ideas

Reinterpretation/RECAST: running new signal samples through analysis, compare to *stored data, background estimates*

Data and Backgrounds are recorded including detector effects. Two approaches to compare to a new signal.

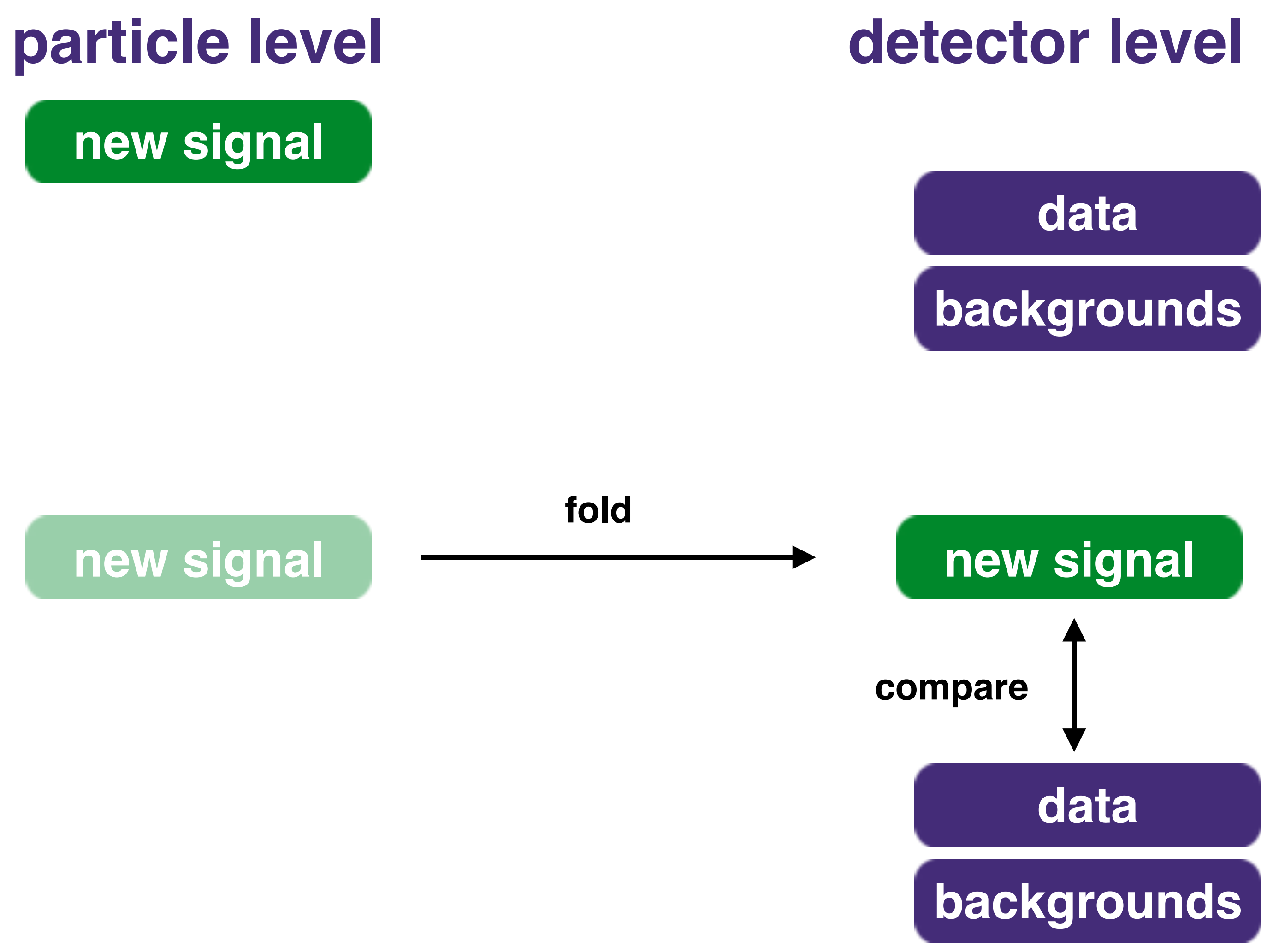
1) unfolded reinterpretation



Reinterpretation/RECAST: running new signal samples through analysis, compare to *stored data, background estimates*

Data and Backgrounds are recorded including detector effects. Two approaches to compare to a new signal.

2) folded reinterpretation



Reinterpretation/RECAST: running new signal samples through analysis, compare to *stored data, background estimates*

Data and Backgrounds are recorded including detector effects. Two approaches to compare to a new signal.

both approaches need an *implementation of the analysis logic:*

- object definitions
- event selection
- statistical analysis

various tools provide infrastructure to cover all or part of the logic

1) unfolded reinterpretation

- Rivet
- MadAnalysis
- homegrown HepMC based analysis

2) folded reinterpretation

- **original analysis by experiment**
- CheckMate
- Rivet (with BSM smearing)
- custom Delphes based analysis
- homegrown toy detectors / approximations

Reinterpretation/RECAST: running new signal samples through analysis, compare to *stored data, background estimates*

Data and Backgrounds are recorded including detector effects. Two approaches to compare to a new signal.

For exotic analyses, unfolded reinterpretations can be difficult. Detector effects are important and non-regular. Not easily described e.g. by a simple convolution; unfolding works best in high-statistics regions, where *folding matrix* can be learnt/inverted well.

1) unfolded reinterpretation

- Rivet
- MadAnalysis
- homegrown HepMC based analysis

2) folded reinterpretation

- **original analysis by experiment**
- CheckMate
- Rivet (with BSM smearing)
- custom Delphes based analysis
- homegrown toy detectors / approximations

How do we ensure, that we can efficiently perform (folded) reinterpretation?

Two solutions serving slightly different purposes

1. **Experiments provide extensive information about analysis details to phenomenologists**
    - allows phenomenologists to *quickly* check a new model.
        - e.g. help you to code your own Delphes/CheckMate/Rivet BSM analysis…
    - broadly asses interesting parameter space regions
    - after information is public (e.g. on HepData) reinterpretation activity independent of experiment

2. **Experiments preserve analyses in full fidelity internally. Re-interpret on-demand for models that are deemed sufficiently promising.**
    - reinterpretation are run *within/by the collaborations,* only public after approval
    - produces an publication-quality, **official** result by the experiments ("gold standard")
    - computationally more involved: e.g. need to run full experiment detector simulation

"Official" reinterpretations (à la 2.) are already being performed, but very manual process. Need to find people that know how to run the analysis, have time to run it, etc. **For Exotics analyses (e.g. LLP) often a problem, because teams are small.**

There is a need for a system that is able to preserve analyses once (at publication time) and re-executes it independent of the original collaboration members.

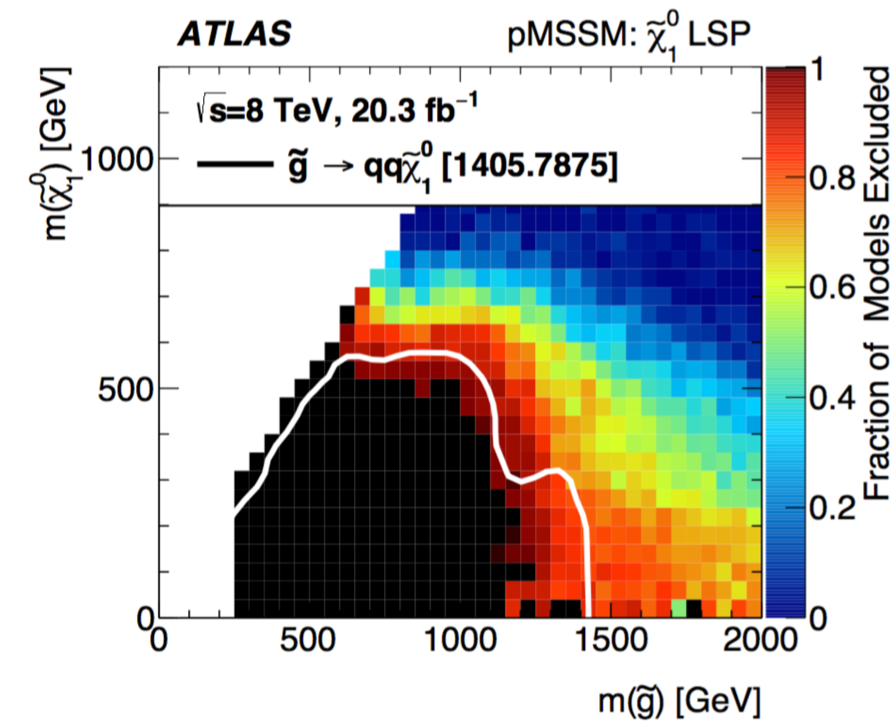
Examples of ‘official’ reinterpretations in ATLAS

JHEP
Published for SISSA by Springer
Received: August 27, 2015
Accepted: September 23, 2015
Published: October 21, 2015

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

arXiv:1508.06608

19-D(!) pMSSM reinterpretation

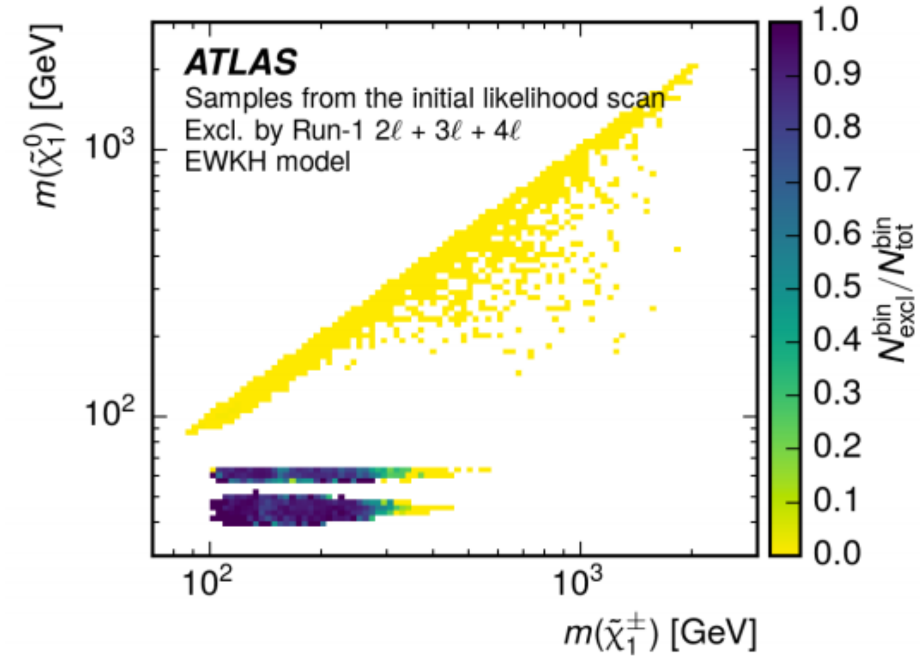


JHEP
Published for SISSA by Springer
Received: August 3, 2016
Accepted: September 22, 2016
Published: September 30, 2016

Dark matter interpretations of ATLAS searches for the electroweak production of supersymmetric particles in $\sqrt{s} = 8$ TeV proton-proton collisions

5-D scan of EWKH sector with help from STA

arXiv:1608.00872



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

RECAST

RECAST is a service that aims to provide the necessary infrastructure for LHC experiments to archive and re-execute their analyses as they were originally designed, without simplifications or approximations. Designed to handle scale and complexity of real-world analyses

Idea:

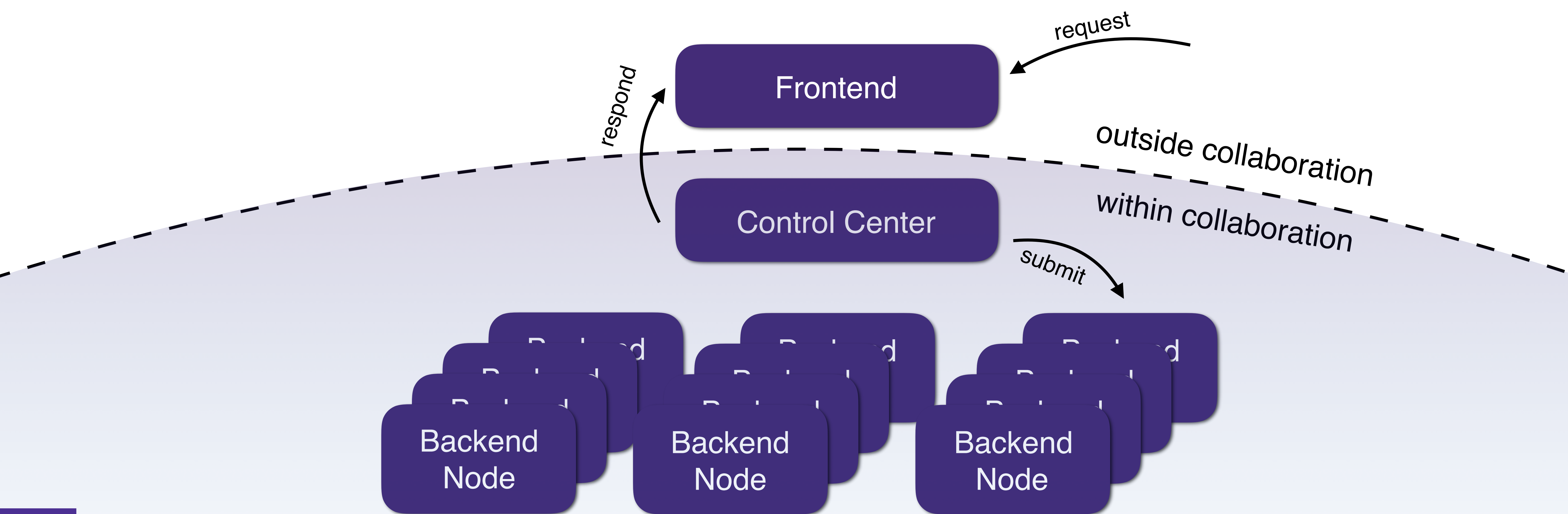
- Allow hep-ph community to *suggest* reinterpretations through a standard (web) interface. They provide most interesting points / scans to do. Auxiliary information such as run cards, SLHA spectra, UFO models
- LHC collaborations review suggestions and choose which to fulfill (based on scale of request, 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

RECAST Infrastructure Overview:

Frontend: public-facing web-service (+ API). Let's e.g. phenomenologists register interest in reinterpretations for specific published analyses. Allows them to provide auxiliary data (SLHA spectrum files, generator run cards, etc..).

Control Center: collaboration-internal web-service (+ API) to inspect incoming requests, compare against catalogue of archived analysis, allows submission to backend to actually perform reinterpretation. Can push "RECAST response" back to frontend.

Backend Cluster: distributed compute resource running on CERN OpenStack infrastructure to execute analysis workflows in order to get reinterpreted result

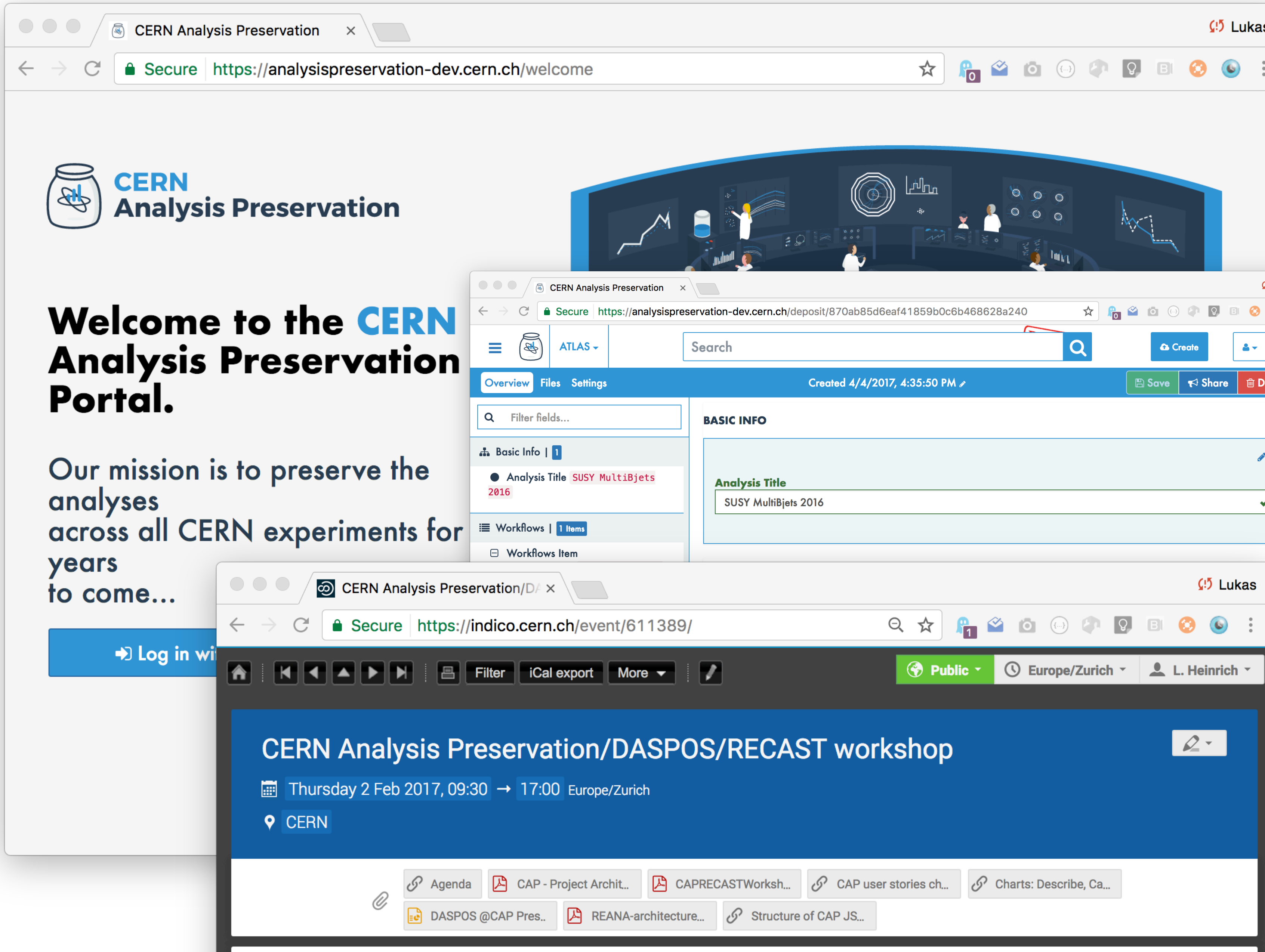


developed in close cooperation with wider HEP software, data preservation and infrastructure efforts, such as CERN Analysis Preservation, diana-hep

DPHEP

DASPOS

dianahep



Reinterpretation Recipe:

At publication time:

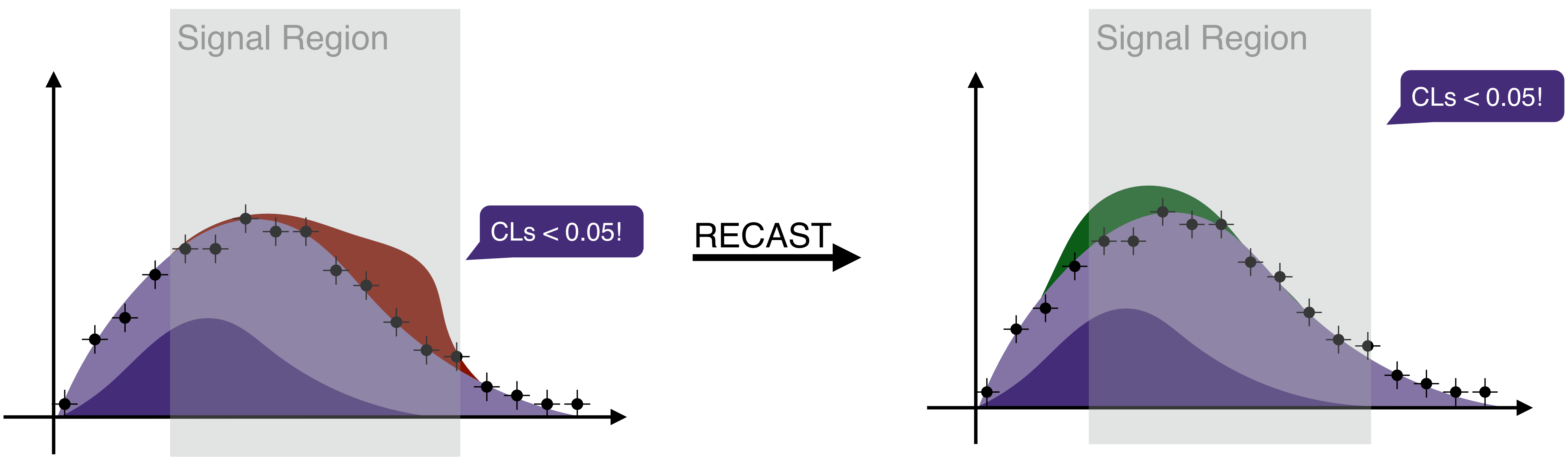
1. archive data + background estimates in fully reduced form, i.e. as histograms (pure HistFactory) or small ntuples (HistFitter). No need to store large upstream background datasets.
2. **preserve original analysis pipeline** (at least such that we can run new signal sample). Not necessary to preserve ability to re-derive background estimates

---

Later:

1. Generate **new signal dataset** with same/compatible settings as original analysis (simulation, reconstruction, etc…)

   after sanity checks, e.g. DM abundance, cross-section, H mass, approx. SR acceptance)
2. Run dataset through **original analysis pipeline,** compare/fit against archived data and backgrounds.



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

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

Reinterpretation Recipe:

Easy OK Hard?

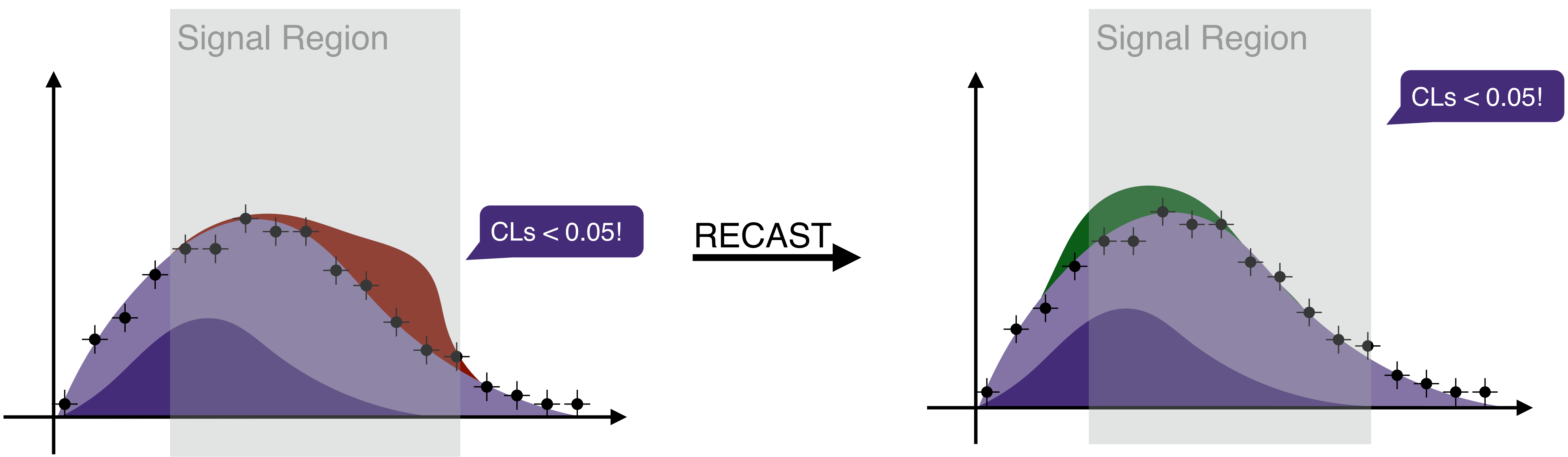
At publication time:

1. archive data + background estimates in fully reduced form, i.e. as histograms (pure HistFactory) or small ntuples (HistFitter)
2. **preserve original analysis pipeline** (at least such that we can run new signal sample). Not necessary to preserve ability to rederive background estimates

---

Later:

1. Generate **new signal dataset** with same/compatible settings as original analysis (simulation, reconstruction, etc…)
   after sanity checks, e.g. DM abundance, cross-section, H mass, approx. SR acceptance)
2. Run dataset through **original analysis pipeline,** compare/fit against archived data and backgrounds.



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

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

Preserving Analyses for RECAST

Analysis Preservation: two-step process

Modern LHC experiment analysis:

- Multiple steps/code-bases, possibly developed by independent teams, with differing software requirements. Example: one team developing the event selection, another team developing the statistical analysis

Need to capture:

1. Individual processing steps

- code bases
- software environments
- identify binaries, scripts in code base
- templates how to run binaries (semantic description of arguments, naming etc..)
- description of step output, what are the relevant data fragments

2. How to connect these steps

- How to wire individual steps together
- What outputs of which steps, are used as inputs for other steps, …

Goal: capture all this with least amount of work for analysis teams, preferably *while analysis is being developed*. *Should not take make than a few days*

Technical Solution:

Preserve Software using industry standard Linux Containers (Docker)

- industry backed (Google, Amazon, …) solution for reproducible software environments. Like a VM, but boots in milliseconds.
- complete freedom for analysis team on software choices. Makes no assumption on how experiments run analysis code.

- can capture conveniently by analysis team:

```
lxplus> docker run …  #start snapshot session
container> svn co …
container> make …
lxplus> docker commit … #save snapshot of workdir
```

Docker Image. Think: executable filesystem snapshot

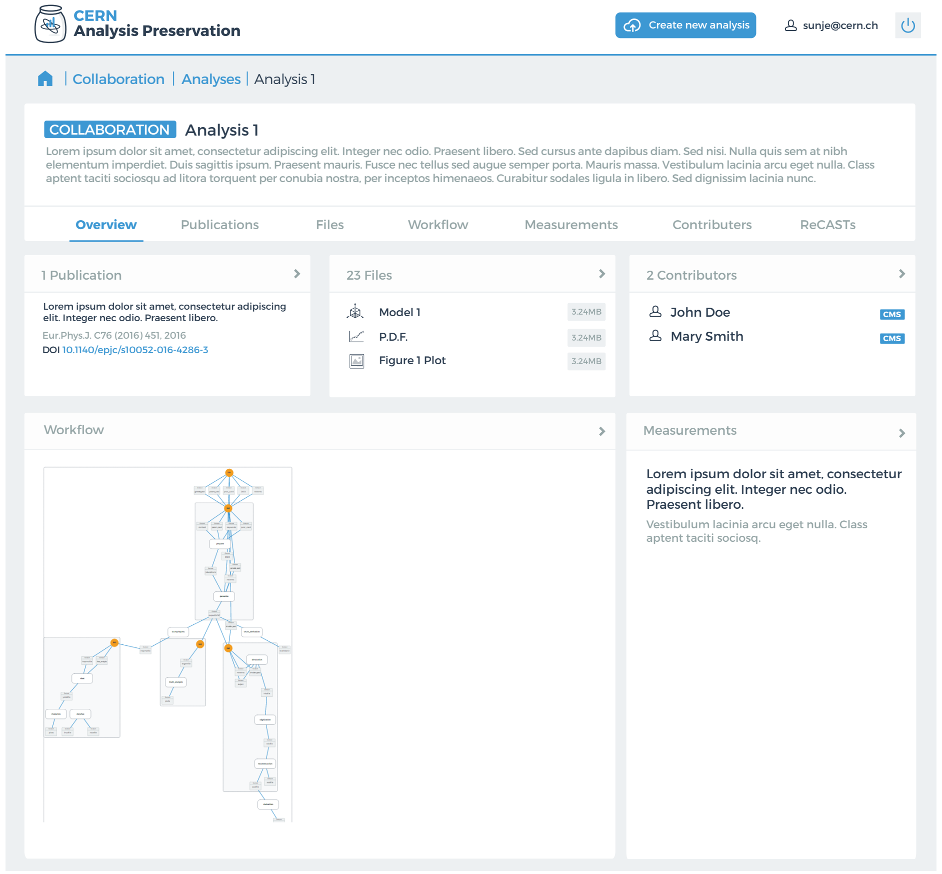
Technical Solution:

Workflow (i.e. logic which steps to run in which order: CxAODMaker → Reader→ Fit)

- in easy to write / read text based format (YAML)
- generic workflow language “**yadage**” based on graphs. No assumption on how you run your analysis. Should be able to accommodate your workflows.
- integrated into CERN Analysis Preservation.
- re-run workflow using tool that interprets info stored in CAP

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

import analysis workflow



CERN Analysis Preservation

Technical Solution:

Thanks to generality of framework, equally applicable to non-experiment analysis (pheno implementation).

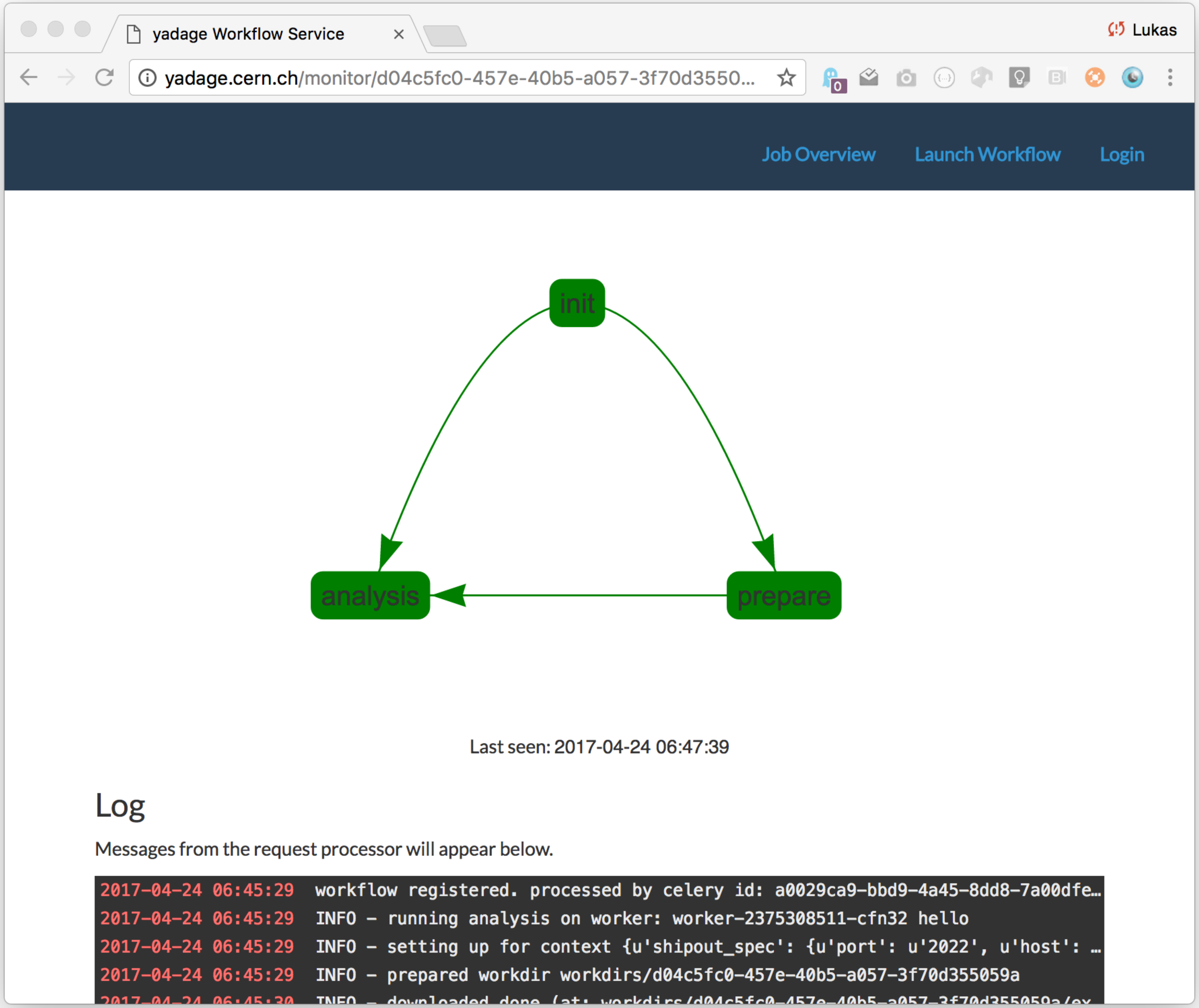
Example courtesy of Giovanna Cottin. Reinterpretation code of arXiv:1606.03099

Workflow:

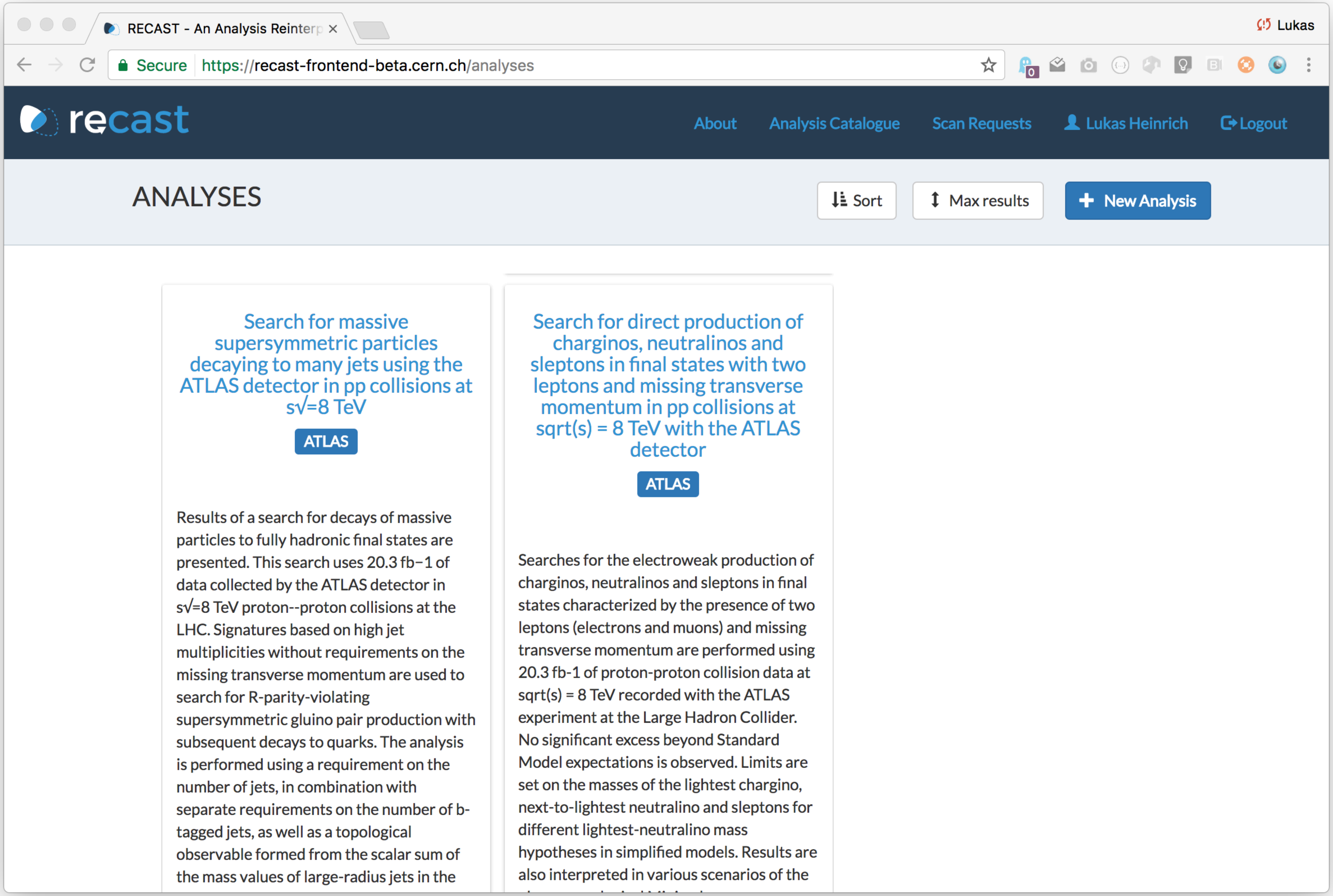
1. **prepare Pythia steering file based on SLHA file**
2. **run Pythia with custom toy ATLAS detector to using steering file outputs: cutflow, signal efficiency**

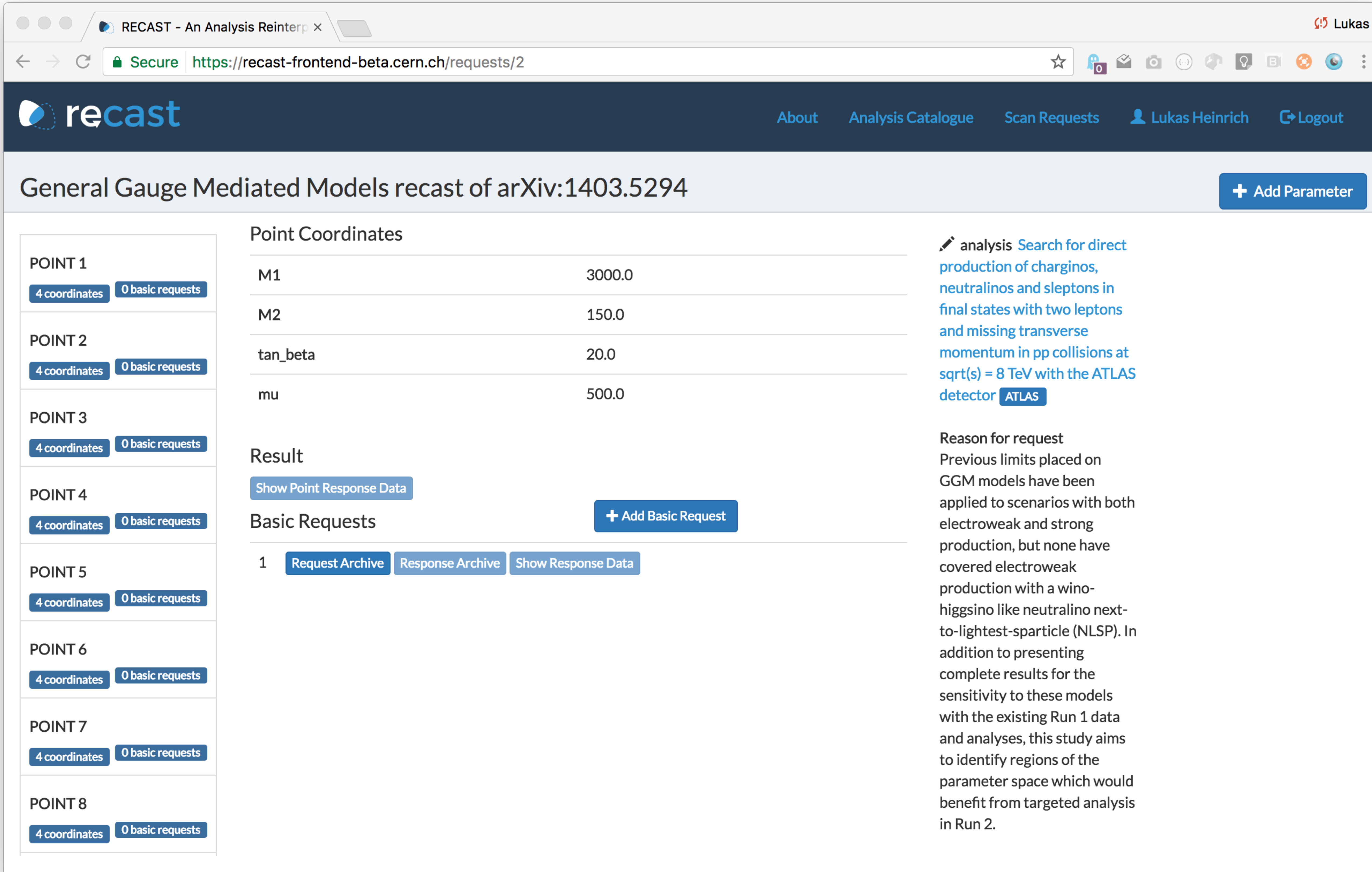
Implemented using same framework as full-fidelity ATLAS analysis

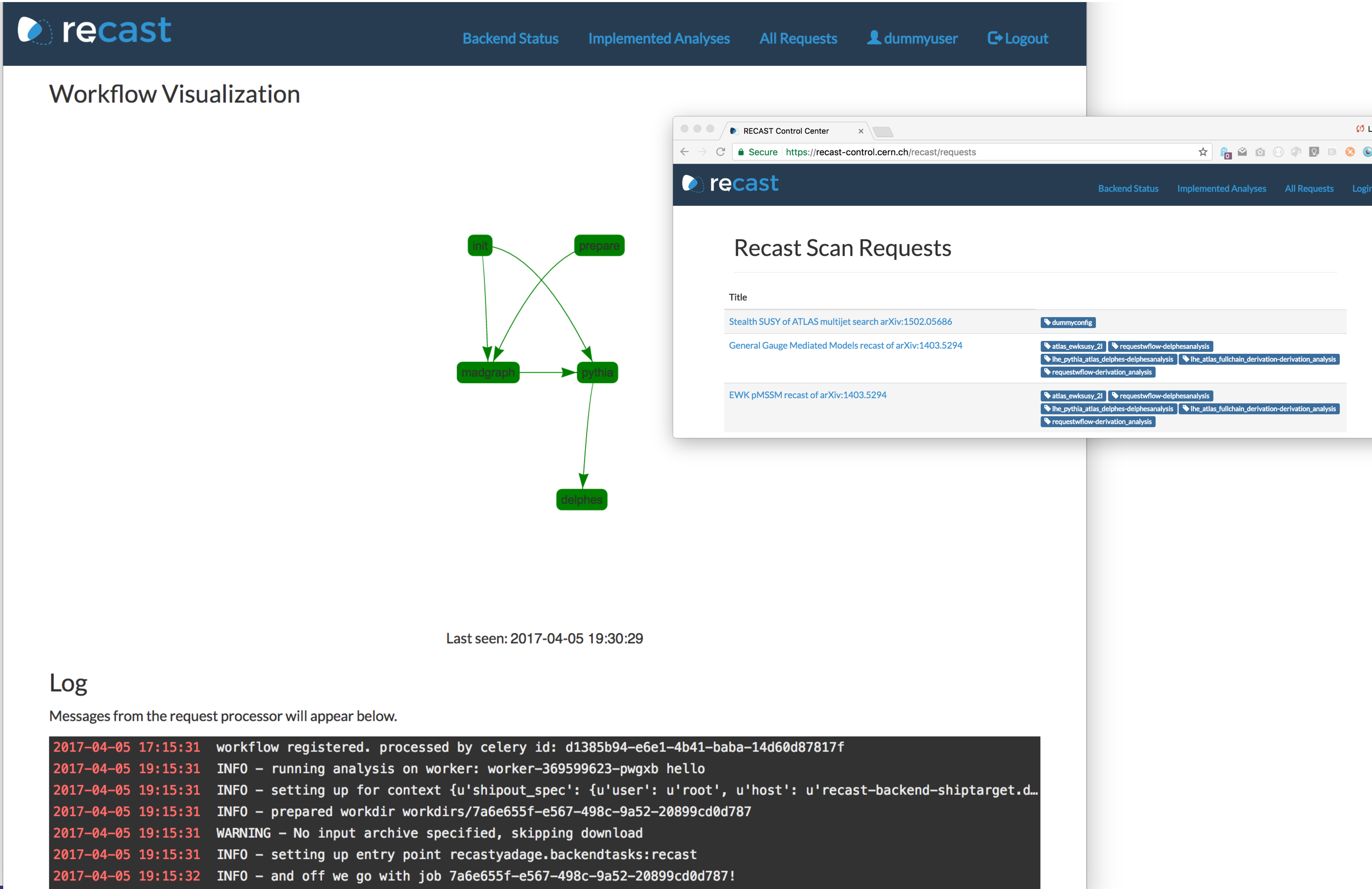
github:lukasheinrich/displaceddemo


yadage Workflow Service
Lukas
yadage.cern.ch/monitor/d04c5fc0-457e-40b5-a057-3f70d3550...
Job Overview
Launch Workflow
Login
init
analysis
prepare
Last seen: 2017-04-24 06:47:39
Log
Messages from the request processor will appear below.
2017-04-24 06:45:29 workflow registered. processed by celery id: a0029ca9-bbd9-4a45-8dd8-7a00dfe…
2017-04-24 06:45:29 INFO - running analysis on worker: worker-2375308511-cfn32 hello
2017-04-24 06:45:29 INFO - setting up for context {u'shipout_spec': {u'port': u'2022', u'host': …
2017-04-24 06:45:29 INFO - prepared workdir workdirs/d04c5fc0-457e-40b5-a057-3f70d355059a

Using RECAST


RECAST - An Analysis Reinterp
Lukas
Secure
https://recast-frontend-beta.cern.ch/analyses
recast
About
Analysis Catalogue
Scan Requests
Lukas Heinrich
Logout
ANALYSES
Sort
Max results
New Analysis
Search for massive supersymmetric particles decaying to many jets using the ATLAS detector in pp collisions at s√=8 TeV
ATLAS
Results of a search for decays of massive particles to fully hadronic final states are presented. This search uses 20.3 fb−1 of data collected by the ATLAS detector in s√=8 TeV proton--proton collisions at the LHC. Signatures based on high jet multiplicities without requirements on the missing transverse momentum are used to search for R-parity-violating supersymmetric gluino pair production with subsequent decays to quarks. The analysis is performed using a requirement on the number of jets, in combination with separate requirements on the number of b-tagged jets, as well as a topological observable formed from the scalar sum of the mass values of large-radius jets in the
Search for direct production of charginos, neutralinos and sleptons in final states with two leptons and missing transverse momentum in pp collisions at sqrt(s) = 8 TeV with the ATLAS detector
ATLAS
Searches for the electroweak production of charginos, neutralinos and sleptons in final states characterized by the presence of two leptons (electrons and muons) and missing transverse momentum are performed using 20.3 fb-1 of proton-proton collision data at sqrt(s) = 8 TeV recorded with the ATLAS experiment at the Large Hadron Collider. No significant excess beyond Standard Model expectations is observed. Limits are set on the masses of the lightest chargino, next-to-lightest neutralino and sleptons for different lightest-neutralino mass hypotheses in simplified models. Results are also interpreted in various scenarios of the


RECAST - An Analysis Reinterp
Lukas
Secure https://recast-frontend-beta.cern.ch/requests/2
recast
About
Analysis Catalogue
Scan Requests
Lukas Heinrich
Logout
General Gauge Mediated Models recast of arXiv:1403.5294
Add Parameter
POINT 1
4 coordinates
0 basic requests
POINT 2
4 coordinates
0 basic requests
POINT 3
4 coordinates
0 basic requests
POINT 4
4 coordinates
0 basic requests
POINT 5
4 coordinates
0 basic requests
POINT 6
4 coordinates
0 basic requests
POINT 7
4 coordinates
0 basic requests
POINT 8
4 coordinates
0 basic requests
Point Coordinates
M1 3000.0
M2 150.0
tan_beta 20.0
mu 500.0
Result
Show Point Response Data
Basic Requests
Add Basic Request
1 Request Archive Response Archive Show Response Data
analysis Search for direct production of charginos, neutralinos and sleptons in final states with two leptons and missing transverse momentum in pp collisions at sqrt(s) = 8 TeV with the ATLAS detector ATLAS
Reason for request
Previous limits placed on GGM models have been applied to scenarios with both electroweak and strong production, but none have covered electroweak production with a wino-higgsino like neutralino next-to-lightest-sparticle (NLSP). In addition to presenting complete results for the sensitivity to these models with the existing Run 1 data and analyses, this study aims to identify regions of the parameter space which would benefit from targeted analysis in Run 2.


recast
Backend Status
Implemented Analyses
All Requests
dummyuser
Logout
Workflow Visualization
init
prepare
madgraph
pythia
delphes
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!
RECAST Control Center
Secure https://recast-control.cern.ch/recast/requests
recast
Backend Status
Implemented Analyses
All Requests
Login
Recast Scan Requests
Title
Stealth SUSY of ATLAS multijet search arXiv:1502.05686
dummyconfig
General Gauge Mediated Models recast of arXiv:1403.5294
atlas_ewksusy_2l
requestwflow-delphesanalysis
lhe_pythia_atlas_delphes-delphesanalysis
lhe_atlas_fullchain_derivation-derivation_analysis
requestwflow-derivation_analysis
EWK pMSSM recast of arXiv:1403.5294
atlas_ewksusy_2l
requestwflow-delphesanalysis
lhe_pythia_atlas_delphes-delphesanalysis
lhe_atlas_fullchain_derivation-derivation_analysis
requestwflow-derivation_analysis

Case Study: Multi B-jets analysis

RECAST - An Analysis Reinterp ×

https://recast-control.cern.ch/yadageresult/result/6/mbj_run2/6

recast — All Requests — lheinric — Logout

Results for processing via Yadage Workflow Plugin for request 6

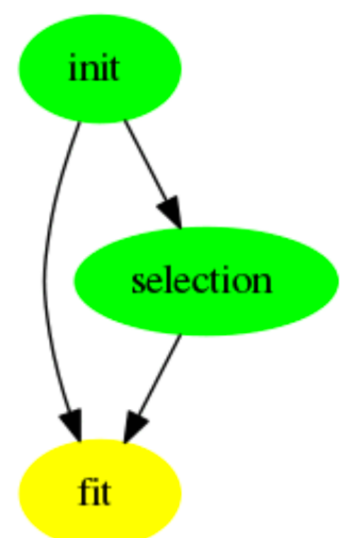
Result Listing

- fitoutput.json
- _adage/adagesnap.txt
- _adage/workflow.gif
- _yadage/yadage_instance_before.json
- _yadage/yadage_instance.json
- _yadage/yadage_template.json
- _yadage/yadage_workflow_instance.dot
- _yadage/yadage_workflow_instance.png
- _yadage/yadage_workflow_instance.pdf

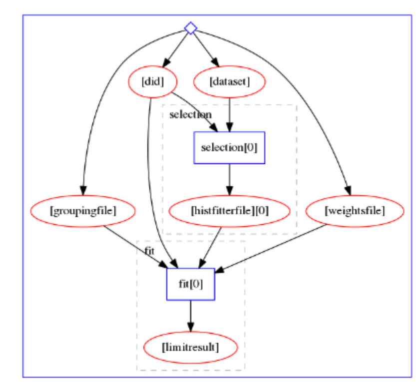
Extracted RECAST Result

-2σ	-1σ	CLs exp.	+1σ	+2σ
1.396691e-06	3.28014e-05	0.0006531772	0.08133995	0.08133995

Workflow Visualization



2016-09-14 11:04:58



How to proceed

The technical problem is largely solved. Able to run large and complex analysis pipelines on scalable computational resources (e.g. 500-1000 CPUs)

Work involved to preserve analysis ~couple of days. Expected to decrease as people become familiar, design analysis with preservation / re-usability in mind

In the next months, we'll focus on getting as many analyses within and outside of the LHC experiments into the framework.

Organizing effort within ATLAS, but happy to work with teams in CMS to get e.g. comparable analysis that look at similar scenarios as ATLAS

Prompt Searches (e.g. standard SUSY searches) are also obvious targets for RECAST and have very wide-applicability. Majority of searches in RECAST will likely be prompt.

Interesting scenarios:

- Reinterpret prompt searches for LLP scenarios to asses where exactly sensitivity end
- Identify multiple LLP searches that are mutually sensitive to their respective signals. Re-use already generated signal datasets and interpret them for another analysis
- Compile a library of custom phono-based codes (such as Giovanna's) associated to published LHC experiments and preserve them such that they are re-usable more widely

Summary

Full analysis preservation / re-execution has traditionally been hard for LHC experiments.

In recent years, specific technological innovations have make it actually feasible.

RECAST infrastructure has been used internally by ATLAS for a number of public results. Ramping up efforts to include more analyses..

LLP scenarios are important use-cases, since full detector simulation is needed, custom reconstructions / object by experiment.

Backend

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

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
```

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

lumi/xsec/KF/FE weighting of HF tree

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}
```

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