

RECAST

Lukas Heinrich

US ATLAS Workshop 2017
Open Session



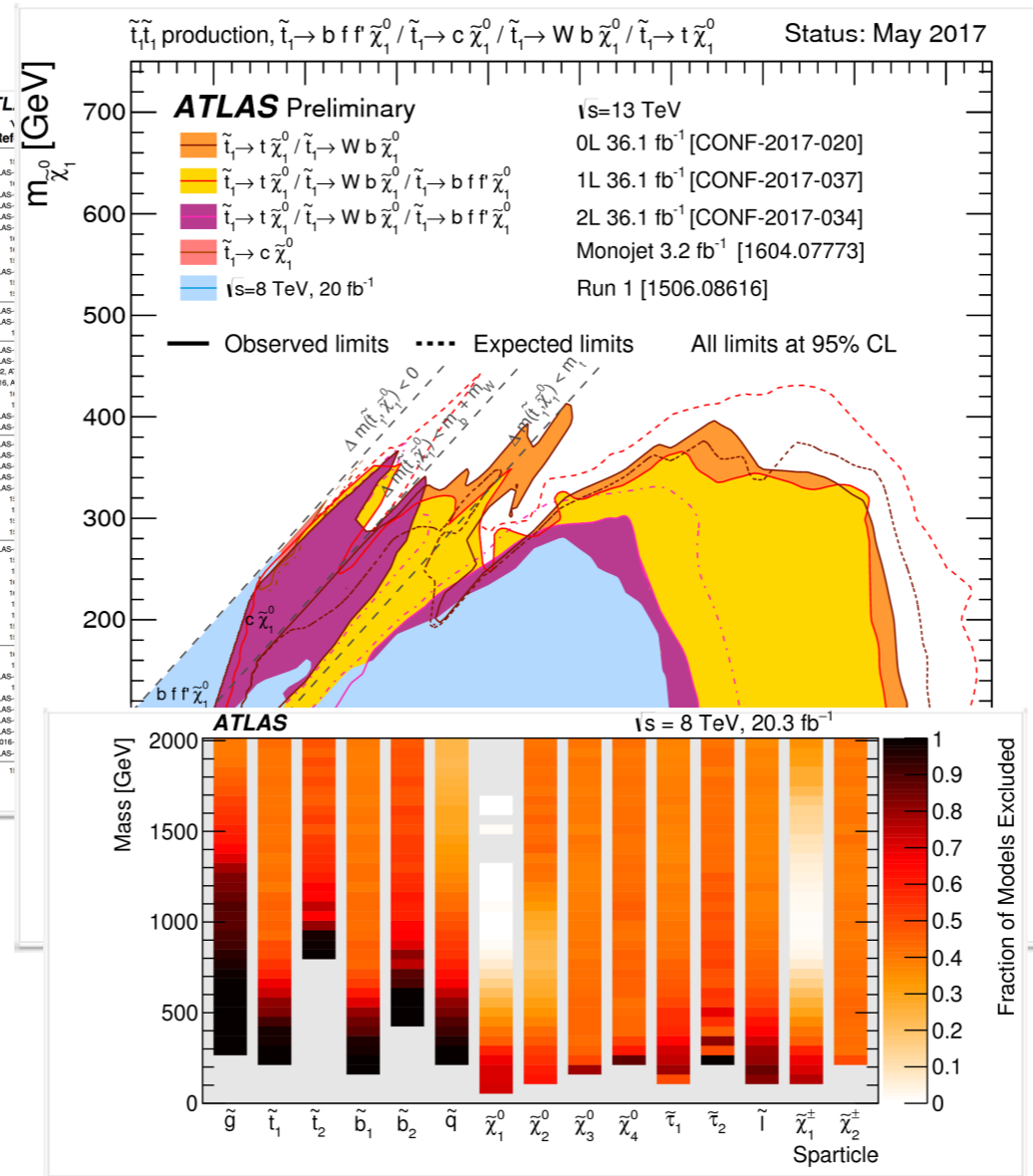
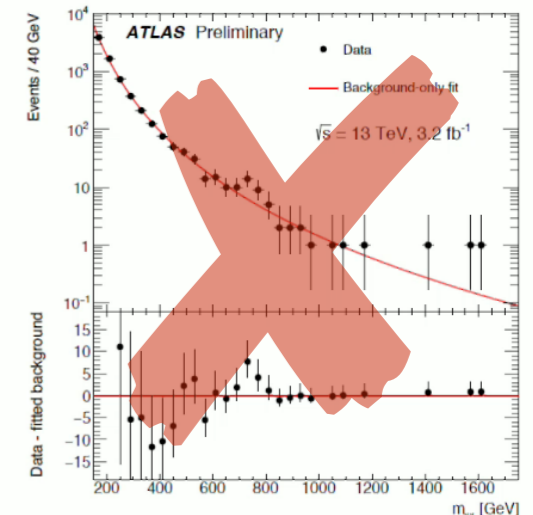
the need for reinterpretation

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



ATLAS SUSY Searches* - 95% CL Lower Limits									
May 2017									
Model	$\epsilon, \mu, \tau, \gamma$	Jets	E_{miss}	$\mathcal{L} d(\text{fb}^{-1})$	Mass limit	$\sqrt{s} = 7, 8 \text{ TeV}$	$\sqrt{s} = 13 \text{ TeV}$	Ref	ATLAS
MSUGRA/CMSSM	0-3 ϵ, μ, τ	2-10 jets	3 b	Yes	20.3	1.85 TeV	1.85 TeV	11	11
$\tilde{g}, \tilde{g} \rightarrow q\bar{q}$	0	2-6 jets	Yes	36.1	1.57 TeV	1.57 TeV	36.1	11	11
$\tilde{g}, \tilde{g} \rightarrow q\bar{q}$ (compressed)	mono-jet	1-3 jets	Yes	3.2	608 GeV	608 GeV	608 GeV	11	11
$\tilde{g}, \tilde{g} \rightarrow q\bar{q}$	0	2-6 jets	Yes	36.1	2.02 TeV	2.02 TeV	2.02 TeV	11	11
$\tilde{g}, \tilde{g} \rightarrow q\bar{q} + \text{gluino}$	0	2-6 jets	Yes	36.1	2.01 TeV	2.01 TeV	2.01 TeV	11	11
$\tilde{g}, \tilde{g} \rightarrow q\bar{q} + \text{gluino} + \tilde{g}$	3 ϵ, μ	4 jets	Yes	36.1	1.825 TeV	1.825 TeV	1.825 TeV	11	11
$\tilde{g}, \tilde{g} \rightarrow q\bar{q} + \text{gluino} + \tilde{g}$	0	7-11 jets	Yes	36.1	1.8 TeV	1.8 TeV	1.8 TeV	11	11
GGM (bino NLSP)	1.2 + 0.1 ϵ	0-2 jets	Yes	3.2	2.0 TeV	2.0 TeV	2.0 TeV	11	11
GGM (bino NLSP)	2 γ	Yes	3.2	1.65 TeV	1.65 TeV	1.65 TeV	1.65 TeV	11	11
GGM (higgsino-bino NLSP)	7	1 b	Yes	20.3	1.37 TeV	1.37 TeV	1.37 TeV	11	11
GGM (higgsino-bino NLSP)	7	2 jets	Yes	13.3	1.8 TeV	1.8 TeV	1.8 TeV	11	11
GGM (higgsino NLSP)	2 ϵ, μ (Z)	2 jets	Yes	20.3	900 GeV	900 GeV	900 GeV	11	11
Gravitino LSP	0	mono-jet	Yes	20.3	865 GeV	865 GeV	865 GeV	11	11
$\tilde{g}, \tilde{g} \rightarrow b\bar{b}$	0	3 b	Yes	36.1	1.92 TeV	1.92 TeV	1.92 TeV	11	11
$\tilde{g}, \tilde{g} \rightarrow b\bar{b}$	0-1 ϵ, μ	3 b	Yes	36.1	1.97 TeV	1.97 TeV	1.97 TeV	11	11
$\tilde{g}, \tilde{g} \rightarrow b\bar{b}$	0-1 ϵ, μ	3 b	Yes	20.1	1.37 TeV	1.37 TeV	1.37 TeV	11	11
$\tilde{g}, \tilde{g} \rightarrow b\bar{b} + \text{gluino}$	0	2 b	Yes	36.1	850 GeV	850 GeV	850 GeV	11	11
$\tilde{g}, \tilde{g} \rightarrow b\bar{b} + \text{gluino}$	2 ϵ, μ (SS)	1 b	Yes	36.1	275-700 GeV	275-700 GeV	275-700 GeV	11	11
$\tilde{g}, \tilde{g} \rightarrow b\bar{b} + \text{gluino}$	0-2 ϵ, μ	1-2 b	Yes	4.7/13.3	117-170 GeV	200-720 GeV	200-720 GeV	11	11
$\tilde{g}, \tilde{g} \rightarrow b\bar{b} + \text{gluino}$	0-2 ϵ, μ	0-2 jets + 1-2 b	Yes	20.3/36.1	90-198 GeV	200-550 GeV	200-550 GeV	11	11
$\tilde{g}, \tilde{g} \rightarrow b\bar{b} + \text{gluino}$	0	mono-jet	Yes	3.2	90-323 GeV	305-550 GeV	305-550 GeV	11	11
$\tilde{g}, \tilde{g} \rightarrow b\bar{b} + \text{gluino}$	2 ϵ, μ (Z)	1 b	Yes	20.3	150-600 GeV	150-600 GeV	150-600 GeV	11	11
$\tilde{g}, \tilde{g} \rightarrow b\bar{b} + \text{gluino}$	3 ϵ, μ (Z)	1 b	Yes	36.1	290-550 GeV	290-550 GeV	290-550 GeV	11	11
$\tilde{g}, \tilde{g} \rightarrow b\bar{b} + \text{gluino}$	1-2 ϵ, μ	4 b	Yes	36.1	325-800 GeV	325-800 GeV	325-800 GeV	11	11
$\tilde{g}, \tilde{g} \rightarrow b\bar{b} + \text{gluino}$	2 ϵ, μ	0	Yes	36.1	90-440 GeV	710 GeV	710 GeV	11	11
$\tilde{g}, \tilde{g} \rightarrow b\bar{b} + \text{gluino}$	2 ϵ, μ	0	Yes	36.1	710 GeV	710 GeV	710 GeV	11	11
$\tilde{g}, \tilde{g} \rightarrow b\bar{b} + \text{gluino}$	2 ϵ, μ	0	Yes	36.1	760 GeV	760 GeV	760 GeV	11	11
$\tilde{g}, \tilde{g} \rightarrow b\bar{b} + \text{gluino}$	3 ϵ, μ	0	Yes	36.1	1.16 TeV	1.16 TeV	1.16 TeV	11	11
$\tilde{g}, \tilde{g} \rightarrow b\bar{b} + \text{gluino}$	2-3 ϵ, μ	0-2 jets	Yes	36.1	580 GeV	580 GeV	580 GeV	11	11
$\tilde{g}, \tilde{g} \rightarrow b\bar{b} + \text{gluino}$	ϵ, μ, γ	0-2 b	Yes	20.3	270 GeV	635 GeV	635 GeV	11	11
$\tilde{g}, \tilde{g} \rightarrow b\bar{b} + \text{gluino}$	4 ϵ, μ	0	Yes	20.3	115-370 GeV	590 GeV	590 GeV	11	11
GGM (wino NLSP) weak prod. $\tilde{g}_1^0 \rightarrow \tilde{g}_2^0$	1 $\epsilon, \mu + \gamma$	Yes	20.3	W	W	W	W	11	11
GGM (bino NLSP) weak prod. $\tilde{g}_1^0 \rightarrow \tilde{g}_2^0$	2 γ	Yes	20.3	W	W	W	W	11	11
Direct $\tilde{g}_1^0 \rightarrow \tilde{g}_2^0$ prod., long-lived \tilde{g}_1^0	Disapp. trk	1 jet	Yes	36.1	430 GeV	430 GeV	430 GeV	11	11
Direct $\tilde{g}_1^0 \rightarrow \tilde{g}_2^0$ prod., long-lived \tilde{g}_1^0	DE/dx trk	Yes	18.4	\tilde{g}_1^0	490 GeV	850 GeV	850 GeV	11	11
Stable, stopped \tilde{g}_1^0 hadron	0	1-5 jets	Yes	27.9	1.98 TeV	1.98 TeV	1.98 TeV	11	11
Stable \tilde{g}_1^0 hadron	trk	-	3.2	1.57 TeV	1.57 TeV	1.57 TeV	1.57 TeV	11	11
Misaligned \tilde{g}_1^0 hadron	DE/dx trk	-	3.2	537 GeV	537 GeV	537 GeV	537 GeV	11	11
GMSB, stable $\tilde{g}_1^0 \rightarrow H, \tilde{g}_1^0 \rightarrow \tau, \mu$	1 μ	-	19.1	440 GeV	440 GeV	440 GeV	440 GeV	11	11
GMSB, $\tilde{g}_1^0 \rightarrow \tilde{g}_2^0$, long-lived \tilde{g}_1^0	2 γ	Yes	20.3	1.0 TeV	1.0 TeV	1.0 TeV	1.0 TeV	11	11
GGM $\tilde{g}_1^0 \rightarrow \tilde{g}_2^0$	disapp. trk	0-2 jets	Yes	20.3	1.0 TeV	1.0 TeV	1.0 TeV	11	11
LFV $\tilde{g}_1^0 \rightarrow \tilde{g}_2^0 + \tilde{g}_1^0 \rightarrow \tau \mu$	disapp. trk	-	3.2	1.9 TeV	1.9 TeV	1.9 TeV	1.9 TeV	11	11
Blinear RPV CMSSM	2 ϵ, μ (SS)	0-3 b	Yes	20.3	1.45 TeV	1.45 TeV	1.45 TeV	11	11
$\tilde{g}_1^0 \rightarrow \tilde{g}_2^0 + \tilde{g}_1^0 \rightarrow \tau \mu$	4 ϵ, μ	Yes	13.3	1.14 TeV	1.14 TeV	1.14 TeV	1.14 TeV	11	11
$\tilde{g}_1^0 \rightarrow \tilde{g}_2^0 + \tilde{g}_1^0 \rightarrow \tau \mu$	3 $\epsilon, \mu + \tau$	Yes	20.3	450 GeV	450 GeV	450 GeV	450 GeV	11	11
$\tilde{g}_1^0 \rightarrow \tilde{g}_2^0 + \tilde{g}_1^0 \rightarrow \tau \mu$	0	4-5 large-R jets	Yes	14.8	1.08 TeV	1.08 TeV	1.08 TeV	11	11
$\tilde{g}_1^0 \rightarrow \tilde{g}_2^0 + \tilde{g}_1^0 \rightarrow \tau \mu$	0	4-5 large-R jets	Yes	14.8	1.85 TeV	1.85 TeV	1.85 TeV	11	11
$\tilde{g}_1^0 \rightarrow \tilde{g}_2^0 + \tilde{g}_1^0 \rightarrow \tau \mu$	1 ϵ, μ	8-10 jets + 0-4 b	Yes	36.1	2.1 TeV	2.1 TeV	2.1 TeV	11	11
$\tilde{g}_1^0 \rightarrow \tilde{g}_2^0 + \tilde{g}_1^0 \rightarrow \tau \mu$	1 ϵ, μ	8-10 jets + 0-4 b	Yes	36.1	1.65 TeV	1.65 TeV	1.65 TeV	11	11
$\tilde{g}_1^0 \rightarrow \tilde{g}_2^0 + \tilde{g}_1^0 \rightarrow \tau \mu$	0	2 jets + 2 b	Yes	15.4	410 GeV	850-510 GeV	850-510 GeV	11	11
$\tilde{g}_1^0 \rightarrow \tilde{g}_2^0 + \tilde{g}_1^0 \rightarrow \tau \mu$	2 ϵ, μ	2 b	Yes	36.1	0.4-1.45 TeV	0.4-1.45 TeV	0.4-1.45 TeV	11	11
Other	Scalar charm, $\tilde{c} \rightarrow c\tilde{g}$	0	2 c	Yes	20.3	510 GeV	510 GeV	11	11

*Only a selection of the available mass limits on new states or phenomena is shown. Many of the limits are based on simplified models, c.f. refs. for the assumptions made.

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

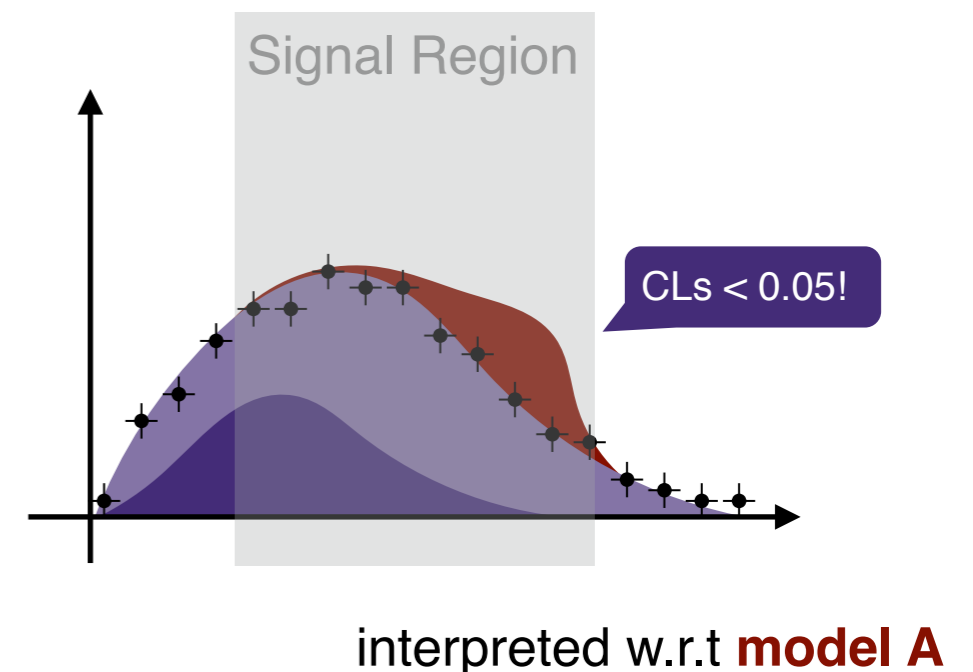
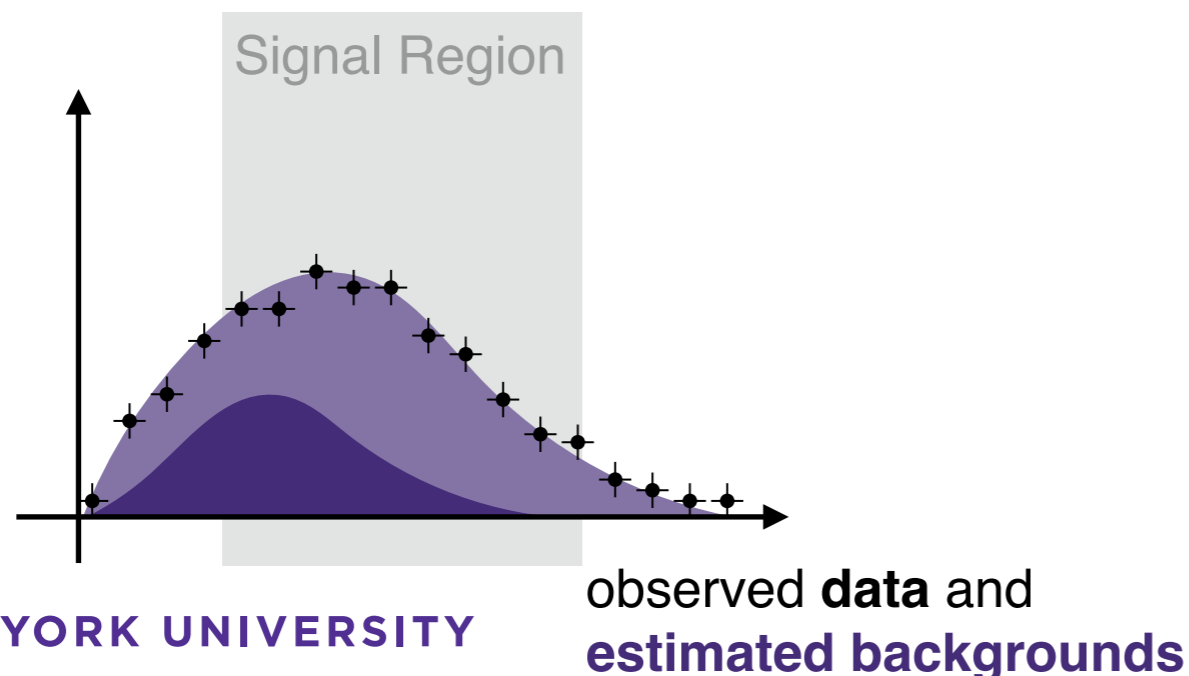
The analyses we prepare at the LHC are **high-effort, expensive projects**: non-trivial amount of person-power, time, and computing resources devoted to achieving a publication-quality result.

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.



the need for reinterpretation

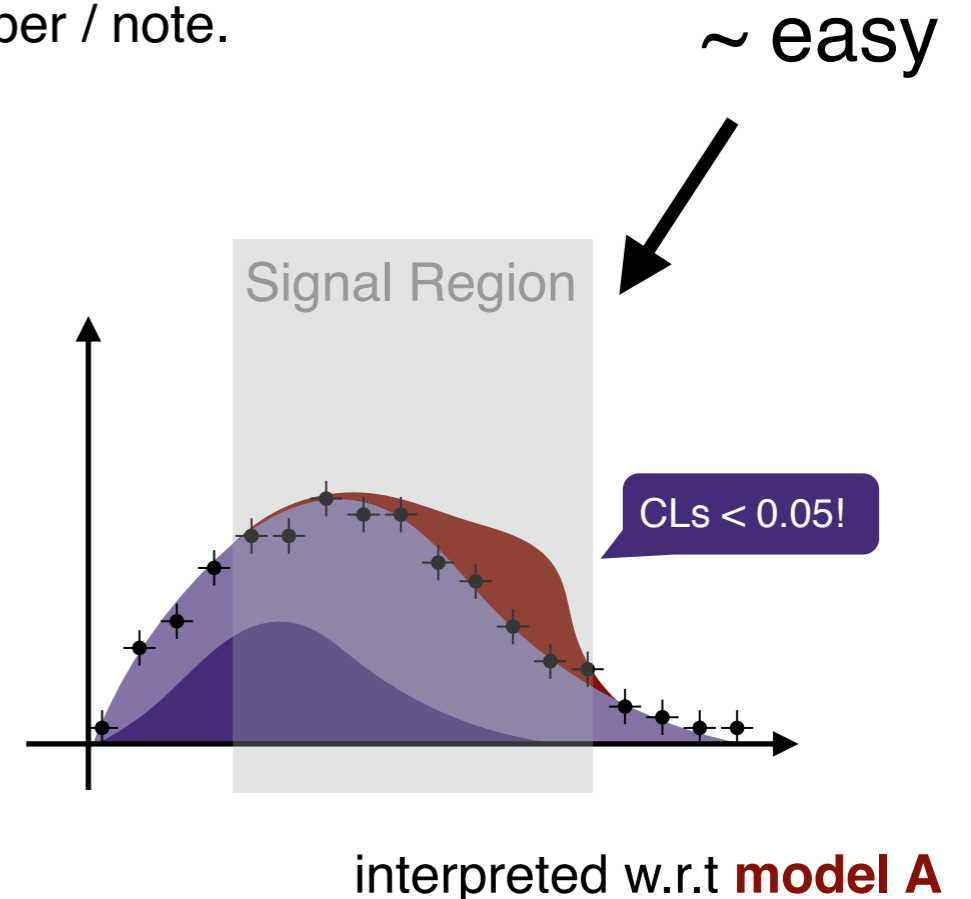
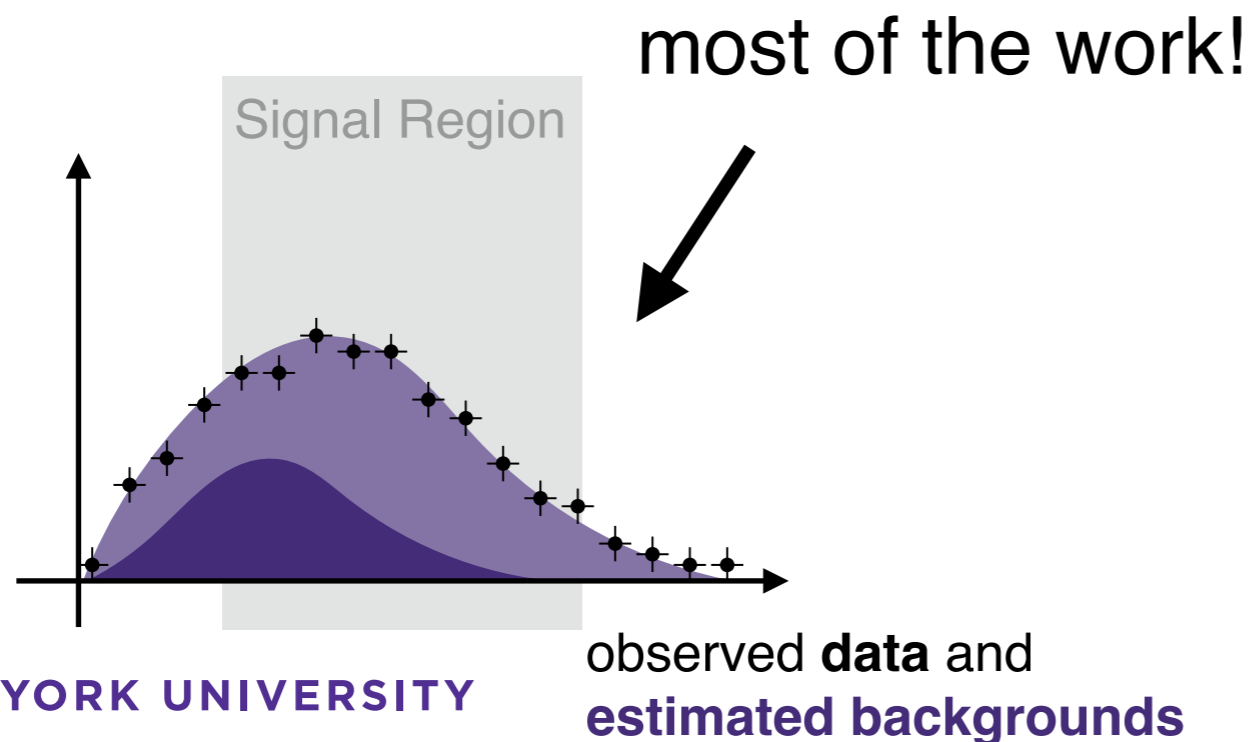
The analyses we prepare at the LHC are **high-effort, expensive projects**: non-trivial amount of person-power, time, and computing resources devoted to achieving a publication-quality result.

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.

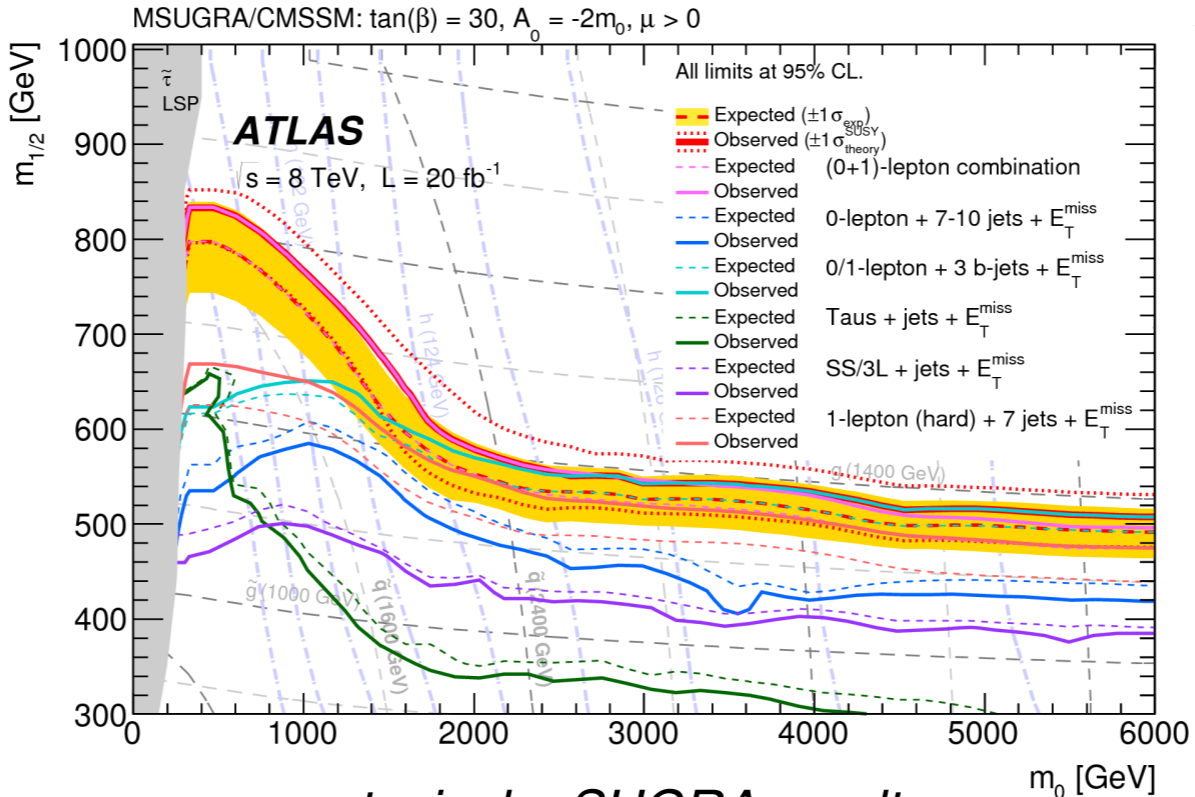
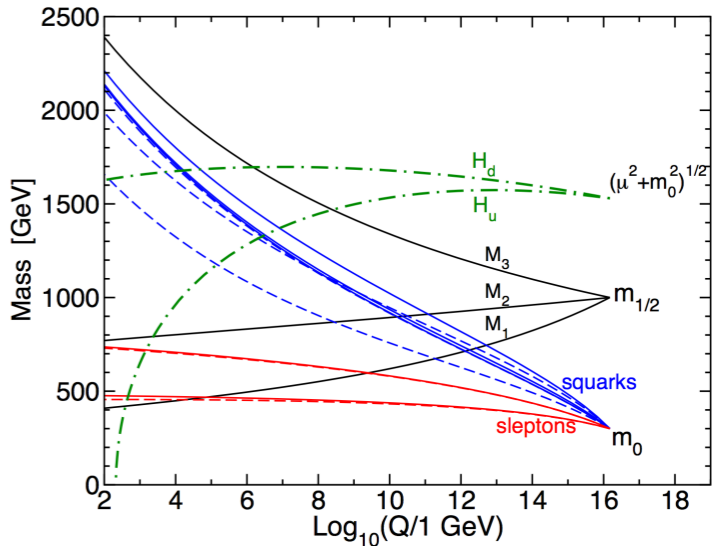


enabling reinterpretation

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 *mSUGRA* 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.

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 \neq strong limit on real model

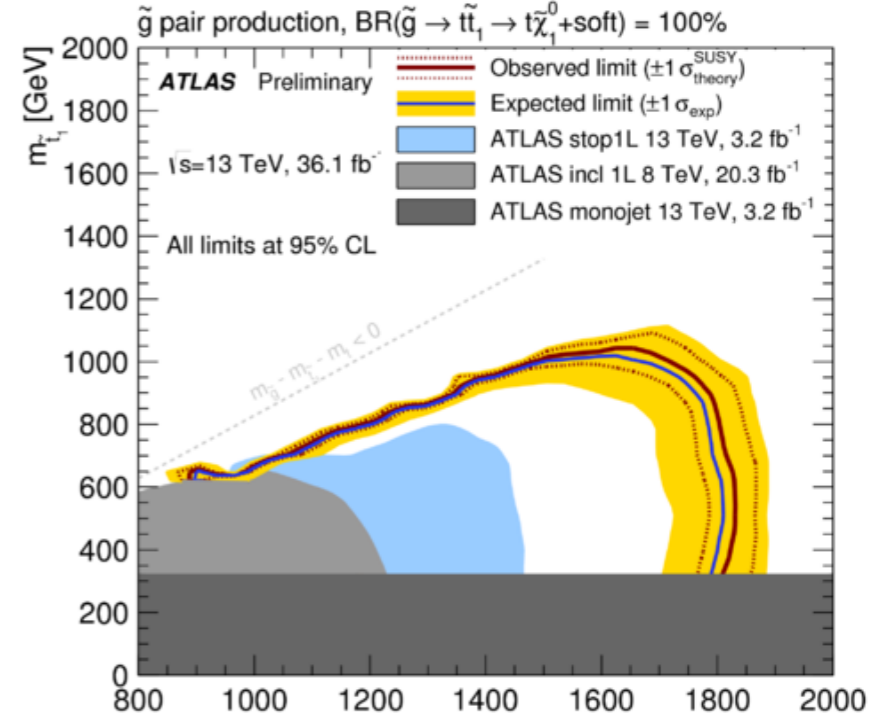
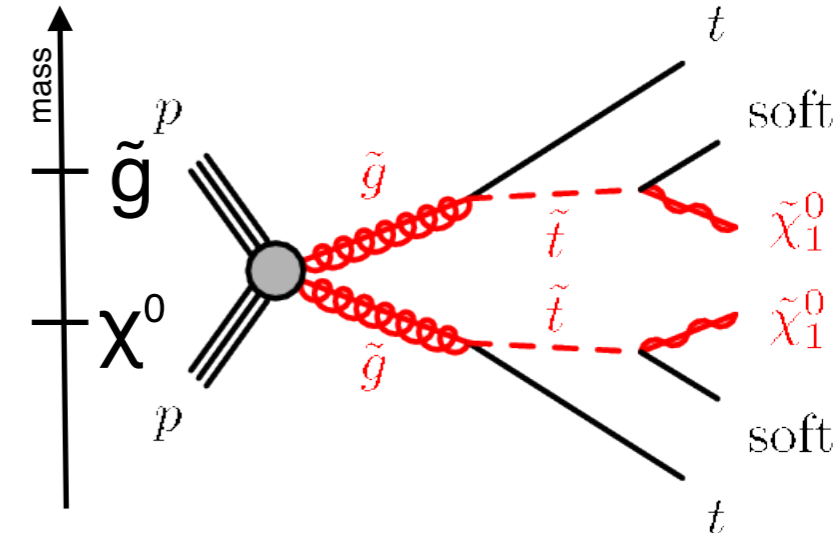
there are still unexcluded natural SUSY models!

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 Zurek²⁶

(LHC New Physics Working Group)

1 [hep-ph] 13 May 2011

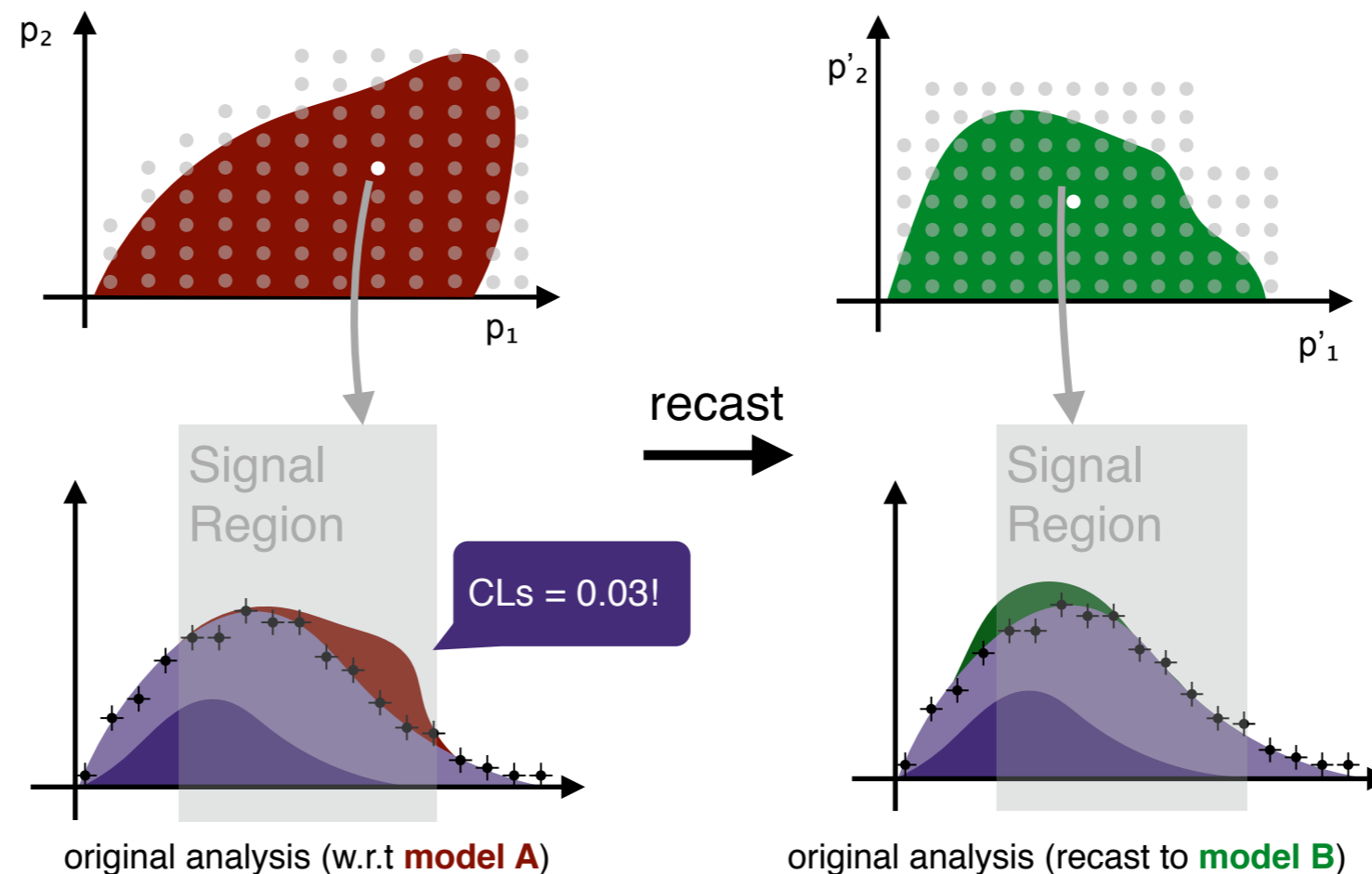


the reinterpretation eco-system

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

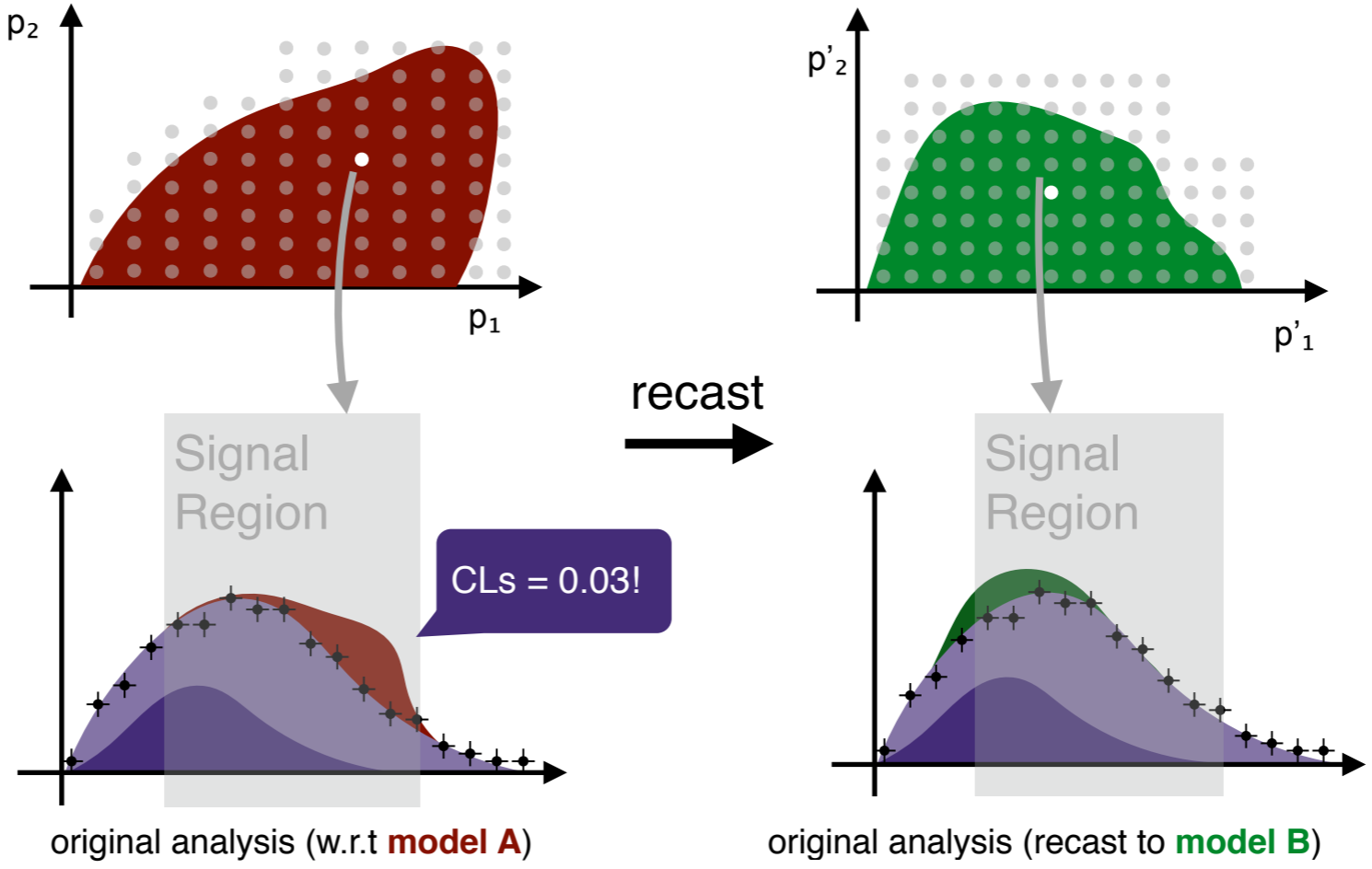
There is a clear need for reinterpretation/recast of analyses to models beyond those matching to simplified models.

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 be done by theorists

can only *really* be done by experiments



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.



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:

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*



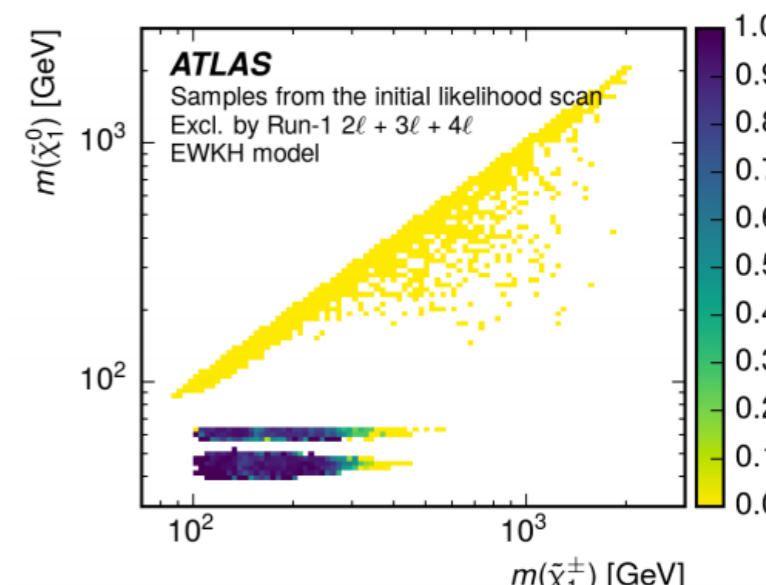
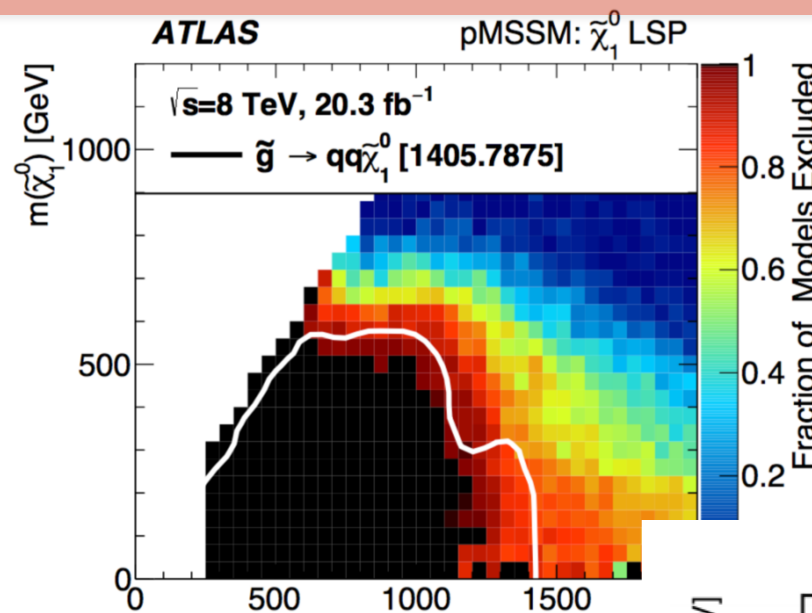
PUBLISHED FOR SISSA BY SPRINGER

arXiv:1508.06608

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

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



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

5-D scan of EWKH sector with help from STA

arXiv:1608.00872



NEW YORK UNIVERSITY

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)

So, what's the hold up?

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



can only *really* be done by experiments

yes, but it's *hard*





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)

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 

can only *really* be done by experiments

yes, but it's *hard*





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)

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

can only *really* be done by experiments

yes, but it's *hard*



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

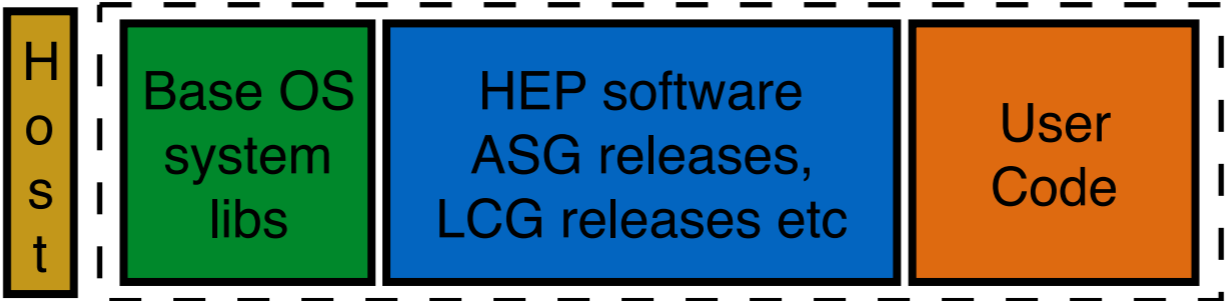
To preserve analyses, we needed to respect the tools, workflows people use. instead of forcing a re-implementation, develop toolchain to capture what they are already doing.

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



ceph



CernVM File system



kubernetes



docker



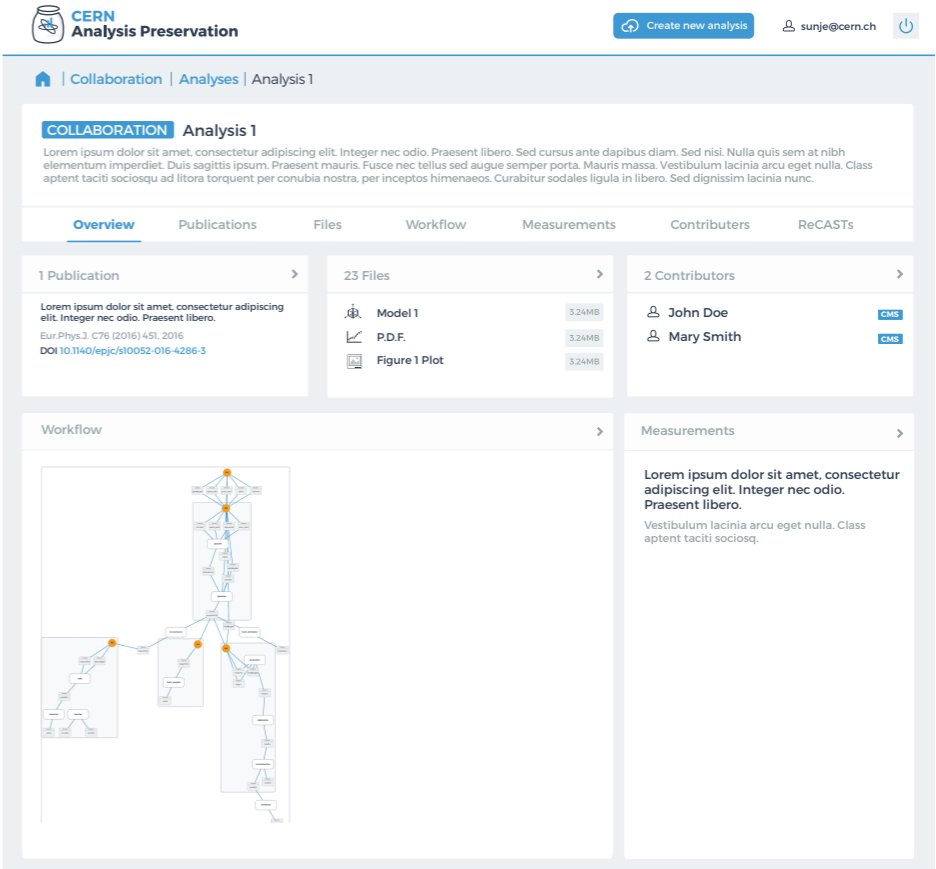
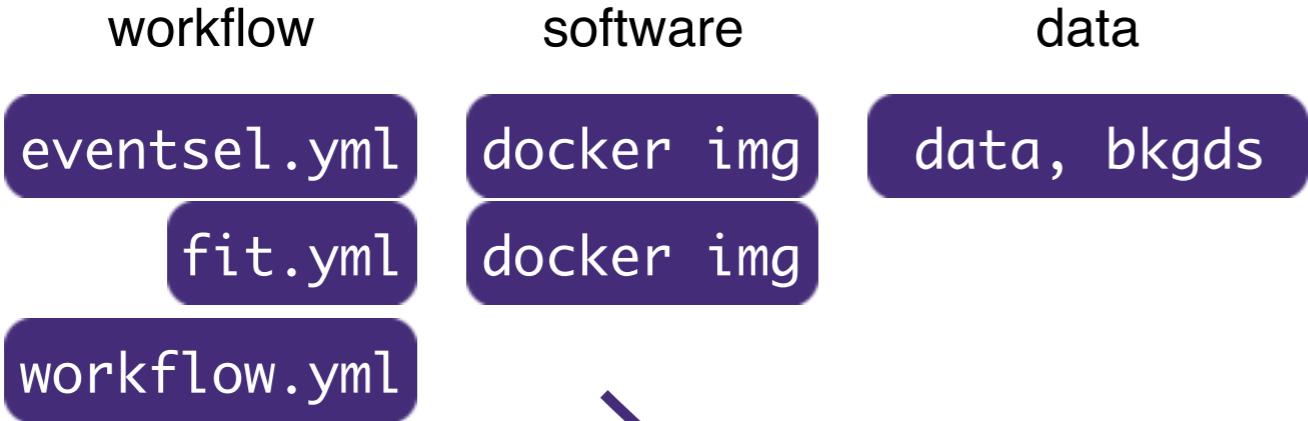
openstack™



Analysis Preservation in ATLAS

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

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.



Analysis Preservation in ATLAS

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

correction:
many more
people involved!

Flip Tanedo @FlipTanedo
Really neat development by @KyleCranmer, @lukasheinrich_, and Diego Rodriguez^2: REusable ANALyses zenodo.org/record/819983#...



1:01 AM - 30 Jun 2017
4 Retweets 10 Likes

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



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.**



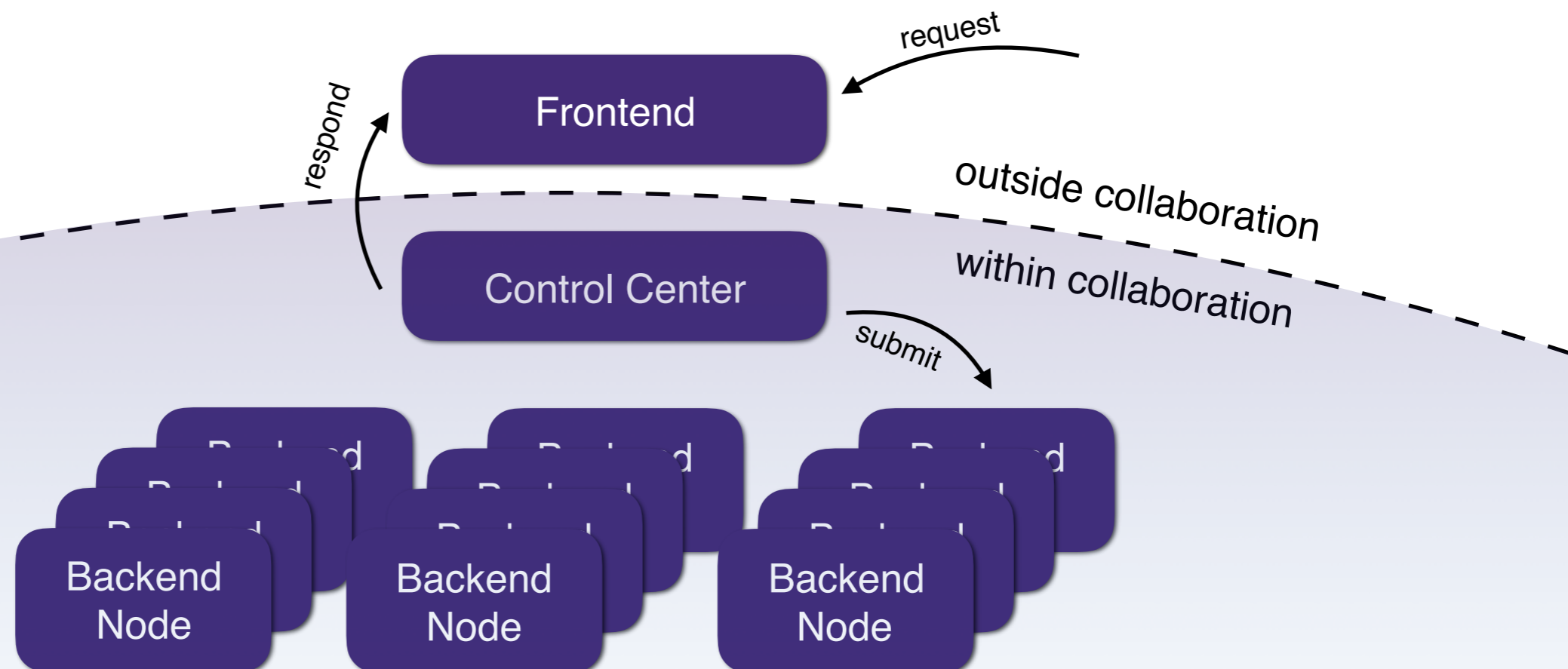
2010! it's been a long time...



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..)

Parameter	Value
M1	3000.0
M2	150.0
tan_beta	20.0
mu	500.0

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 Control Center Lukas

Not Secure <https://localhost:8000/recast/request/10>

recast Backend Status Implemented Analyses All Requests Login

Recast Scan Request A three-dimensional RECAST

Request Details

© 2017-07-27

Analysis

Search for massive supersymmetric particles decaying to many jets using the ATLAS detector in pp collisions at $\sqrt{s}=8$ TeV

Reason:

the original analysis only investigates simplified models. The branching ratio of this particular complete model into the studied topologies is quite small, so it would be interesting to understand what the true exlusions are.

Additional Info:

the most interesting points are the ones attached. Possibly this scan could be extended in case the sensitivity is higher than expected.

Process All ▾

Requested Parameter

Parameter Point 1

collaboration internal dashboard

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

recast Backend Status Implemented Analyses All Requests dummy

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'chipout_spec': {u'user': u'root', u'host': u'recast-backen

```

RECAST - An Analysis Reinter... All Requests lheinric Logout

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

recast

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

Workflow Visualization

Extracted RECAST Result

-2 σ	-1 σ	CLs exp.	+1 σ	+2 σ	
1.396691e-06	3.28014e-05	0.0006531772	0.08133995	0.08133995	2016-09-14 11:04:58



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 by the 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 a 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

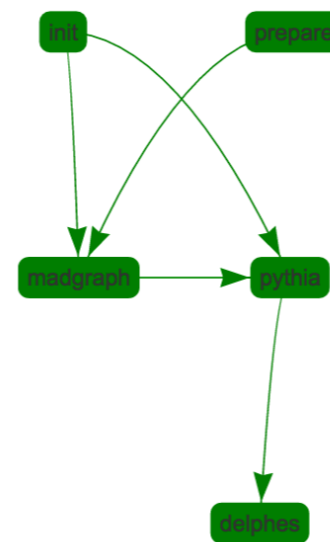
Happy to organize 1-/2-day workshop/hackathon to work with multiple analyses in unison.



Appendix



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..)

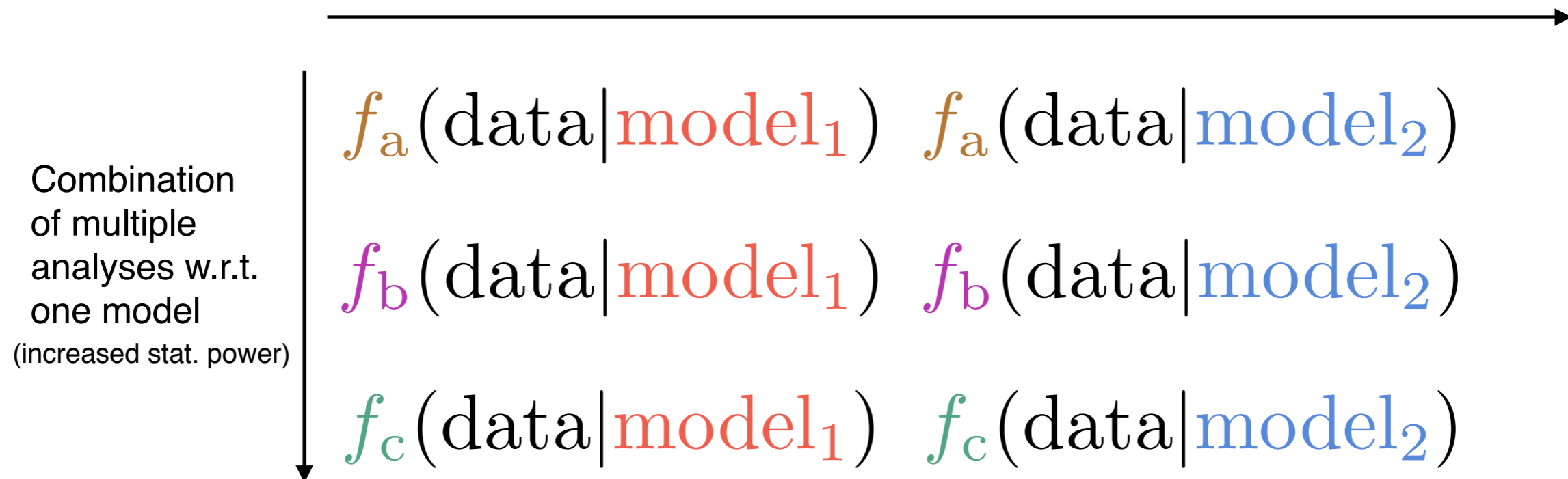


archive analysis in a **parametrized form**, such that we can quickly run a new model

$$f_{\text{analysis}}(\text{data} | \cdot)$$

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



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
1 process:
2 process_type: 'interpolated-script-cmd'
3 script: |
4 #!/bin/bash
5 echo "Hello"
6 source ~/.bashrc
7 setupATLAS
8 source ./rcSetup.sh
9 /recast_auth/getmyproxy.sh
10 lsetup fax dq2
11 python MultibjetsAnalysis/scripts/Run.py --dataSource 1 --doSyst 1 --doNTUPSyst 1 --doNTUP 0 --doxAOD 0 --doH
12 mv {submitdir}/data-output_histfitter/*.root {outputprefix}.{did}.root
13 publisher:
14 publisher_type: 'fromglob-pub'
15 globexpression: '*.root'
16 outputkey: histfitterfile
17 environment:
18 environment_type: 'docker-encapsulated'
19 image: lukasheinrich/multibsel_cvmfs
20 resources:
21 - CVMFS
22 - GRIDProxy
```

```
49 lines (42 sloc) | 1.42 KB
1 process:
2 process_type: 'interpolated-script-cmd'
3 script: |
4 #!/bin/bash
5 source ~/.bashrc
6 setupATLAS
7 lsetup "root 6.06.02-x86_64-slc6-gcc48-opt"
8 cd /code/multib/HistFitter
9 source ./setup.sh
10 cd analysis/analysis_multib
11
12
13 /recast_auth/getkrb.sh
14 #klist
15 #exit
16 cd input
17 python mergeTrees.py {selectionoutput} --filters filters/filters_ht.json --weights {weightsfile} --did-to-group {groupingfi
18 cd ..
19
20 lumi="5807.51"
21 grid="{gridname}"
22 region="Gbb_A"
23 tag="tag2.4.11-1-0_July00"
24 echo '{"pointname}">' > point.json
25 cat point.json
26 export HF_MBJ_SIGNALJSON="point.json"
27 export HF_MBJ_BACKGROUNDFILE={bkgtree}
28 export HF_MBJ_DATAFILE={datatree}
29 export HF_MBJ_SIGNALFILE='input/Sig.root'
30 HistFitter.py -wtpf -F excl python/My3bGtt.py _signalRegion $region _lumi $lumi _unblind true _doHFSplitting false 2>&1 | t
31
32 resultfile=$(ls results/My3bGtt_*fixSigXSecNominal*_hypotest.root)
33 echo "result file is: $resultfile"
34 root -b -q 'root2json.C("'"$resultfile"'", "hypo_Gbb_%f_%f")'
35
36 jsonfile=$(ls *harvest_list.json)
37 python recast_format.py $jsonfile {outputjson}
38
```

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

Extract Results into JSON format

Run HF



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

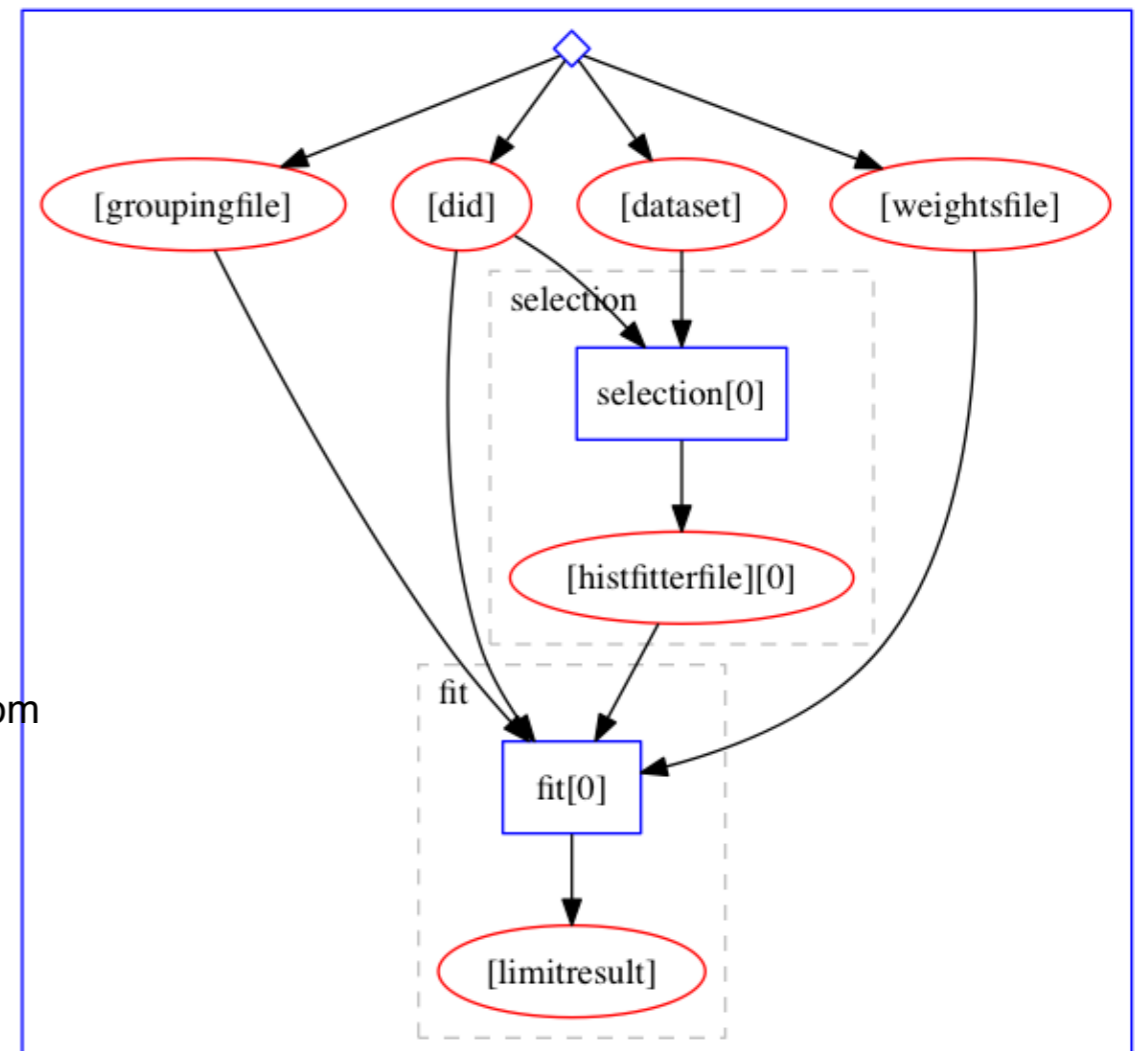
27 lines (26 sloc) | 1.07 KB

Raw Blame History

```
1 stages:
2   - name: selection
3     dependencies: ['init']
4     scheduler:
5       scheduler_type: singlestep-stage
6       parameters:
7         dataset: {stages: init, output: dataset, unwrap: true}
8         submitdir: '{workdir}/submitdir'
9         outputprefix: '{workdir}/histfitter.root'
10        did: {stages: init, output: did, unwrap: true}
11        step: {$ref: 'selscript.yml#'}
12   - name: fit
13     dependencies: ['selection']
14     scheduler:
15       scheduler_type: singlestep-stage
16       parameters:
17         bkgtree: 'root://eosuser.cern.ch///eos/project/r/recast/Bkg_2.4.15-2-0_merged.root'
18         datatree: 'root://eosuser.cern.ch///eos/project/r/recast/Data_2.4.15-2-0.root'
19         outputjson: '{workdir}/fitoutput.json'
20         pointname: 'Gbb_1600_200'
21         gridname: Gbb
22         selectionoutput: {stages: selection, output: histfitterfile, unwrap: true}
23         weightsfile: {stages: init, output: weightsfile, unwrap: true}
24         groupingfile: {stages: init, output: groupingfile, unwrap: true}
25         did: {stages: init, output: did, unwrap: true}
26         step: {$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)

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

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>”

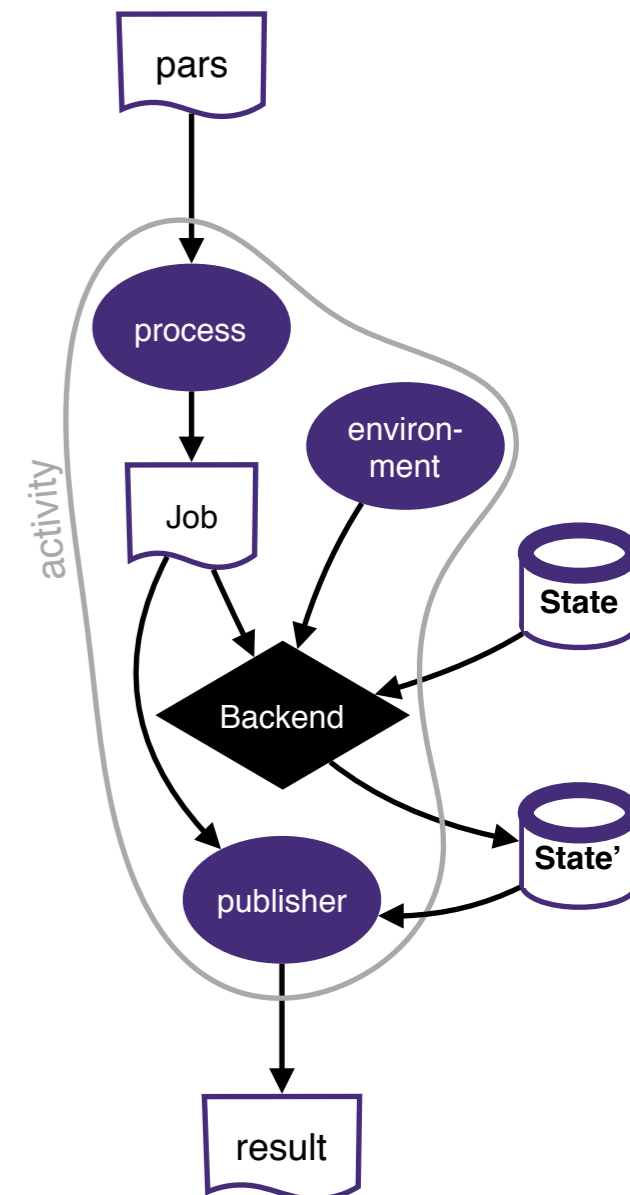
concrete: “./DelphesHepMC /input/file/path.hepmc /output/file.root”

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

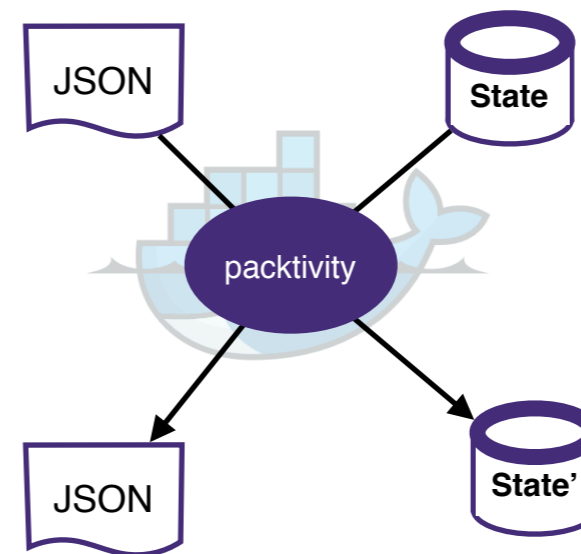
- as interchange format for parameters and result data
- as declarative description format for *process/env/publisher*
 - incl. JSON schemas for validation

Essentially, a self-consistent “packaged activity” – a “packtivity”

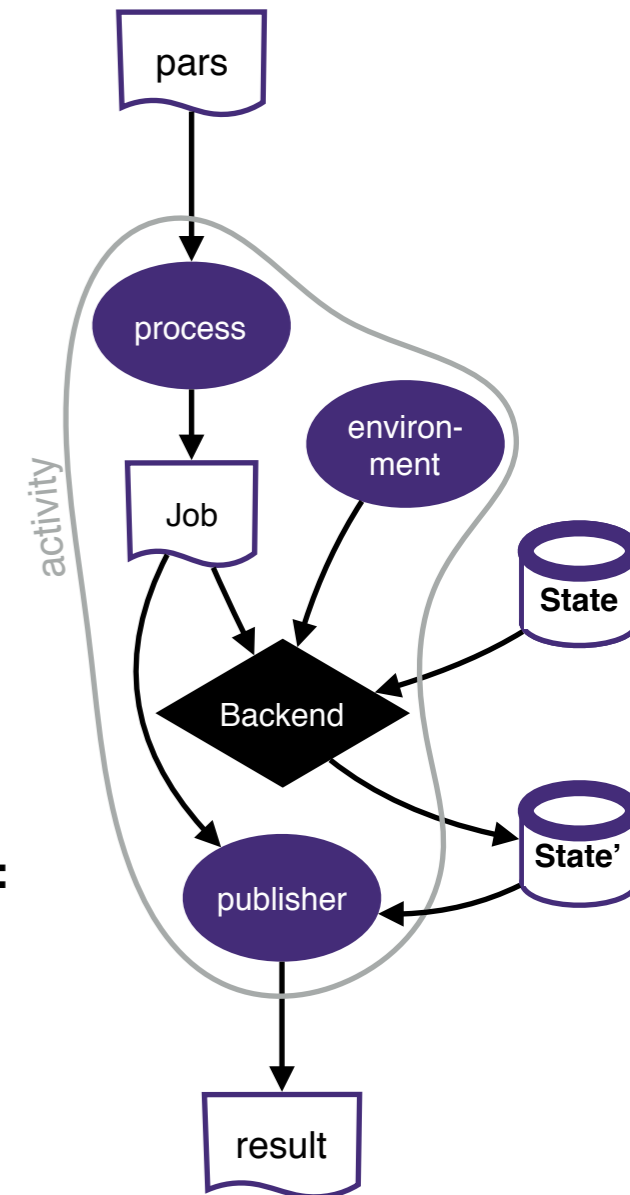
- JSON API
- archivable, declarative description as JSON
- dependencies captured in environment
 - e.g. Docker Image

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
```

python package: “packtivity”

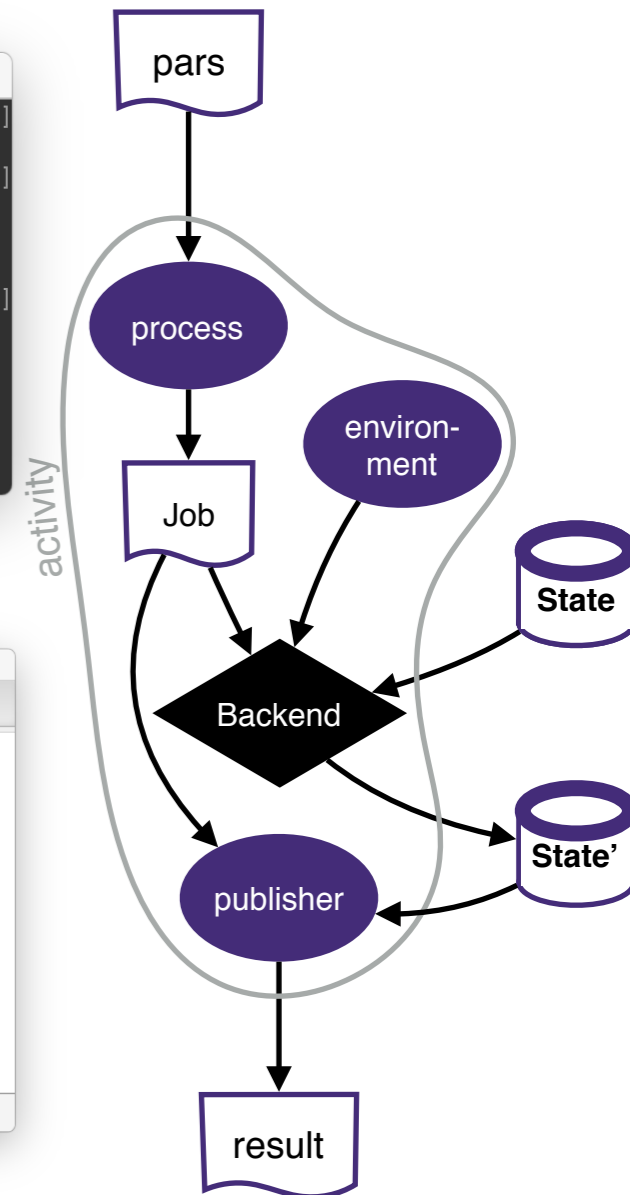
- executes packtivities according to JSON spec for given parameters
- cli tool and python bindings
- multi-host / remote execution ready via e.g. Docker Swarm

CLI tool

```
172-27-219-223 — -zsh — 64x12
[$> ls
delphes.yml input.hepmc pars.yml
[$> pygmentize -g pars.yml
delphes_card: 'delphes/cards/delphes_card_ATLAS.tcl'
inputhepmc: '{workdir}/input.hepmc'
outputroot: '{workdir}/out.root'
[$> packtivity-run delphes.yml pars.yml
{'rootfile': '/Users/lukas/chep2016/out.root'} (prepublished)
[$>
```

python bindings

```
pack.py — /Users/lukas/chep2016
pack.py
1 import os
2 import capschemas
3 from packtivity import packtivity
4 from packtivity.statecontexts.poxisfs_context import make_new_context
5 packtivity_description = capschemas.load(
6     'delphes.yml', os.getcwd(),
7     'packtivity/packtivity-schema')
8 pars = {
9     delphes_card: 'delphes/cards/delphes_card_ATLAS.tcl',
10    inputhepmc: '{workdir}/input.hepmc',
11    outputroot: '{workdir}/out.root'
12 }
13 packtivity(packtivity_description, parameters, make_new_context(os.getcwd()))
```



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

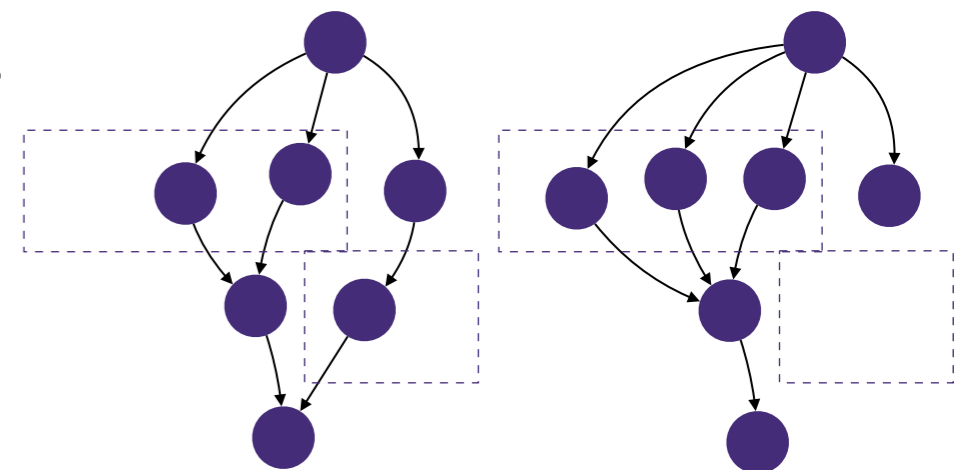
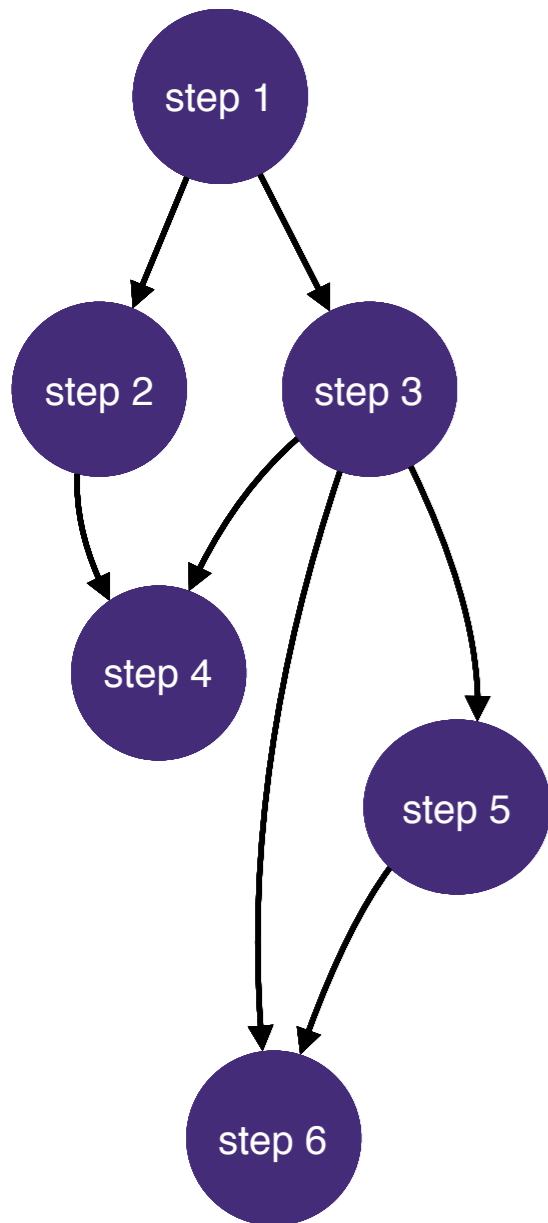
2. Problem: Preserve Parametrized Workflow

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

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

Two place where parametrization enter:

1. individual steps parametrized: covered by “packactivities”
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



Par. Set 1

Par. Set 2



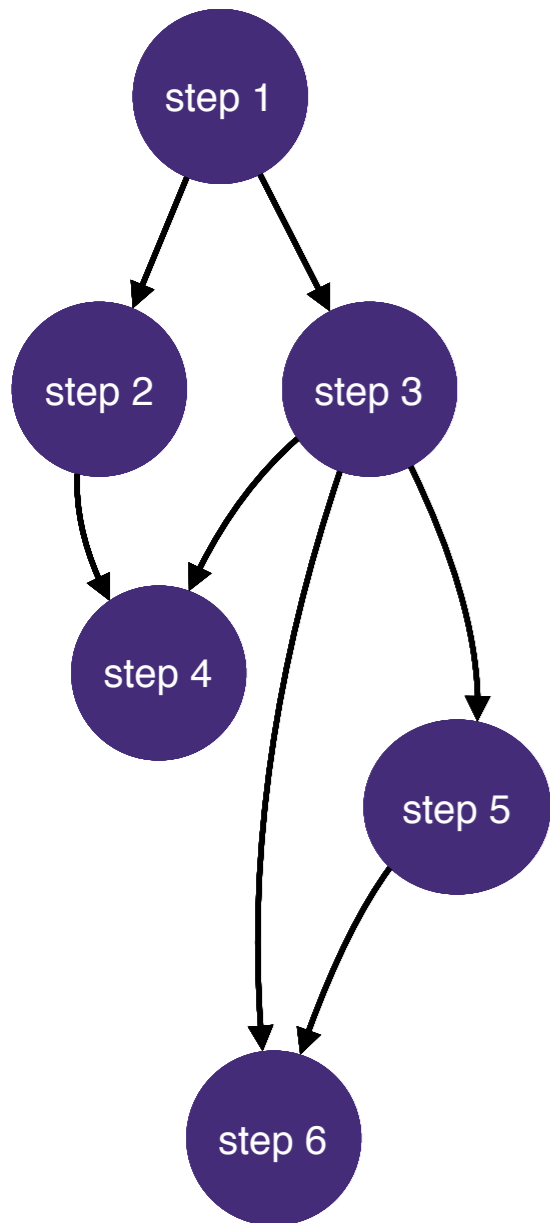
How to preserve $f_{\text{analysis}}(\cdot)$?

2. Problem: Preserve Parametrized Workflow

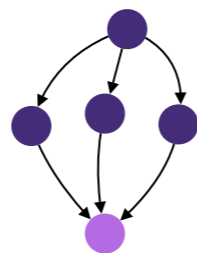
Therefore: Sequentially build up graph, as sufficient information becomes available, using a number of stages that add nodes and edges

To capture analysis workflow, capture the stages.

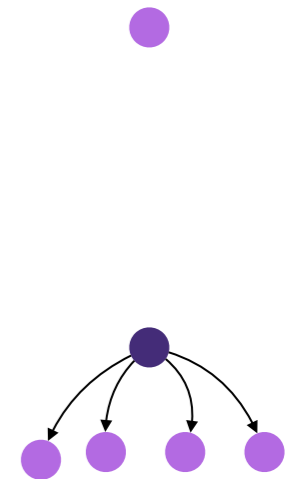
**Example:
Parametrized
Map-Reduce**



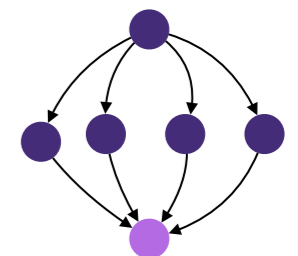
Stage 1:
unknown number of files. e.g.
download & unpack archive with a
priori unknown # of files



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



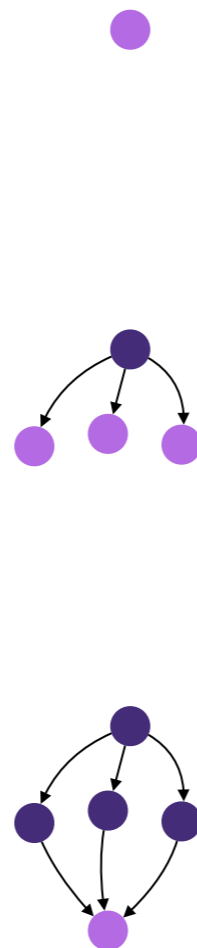
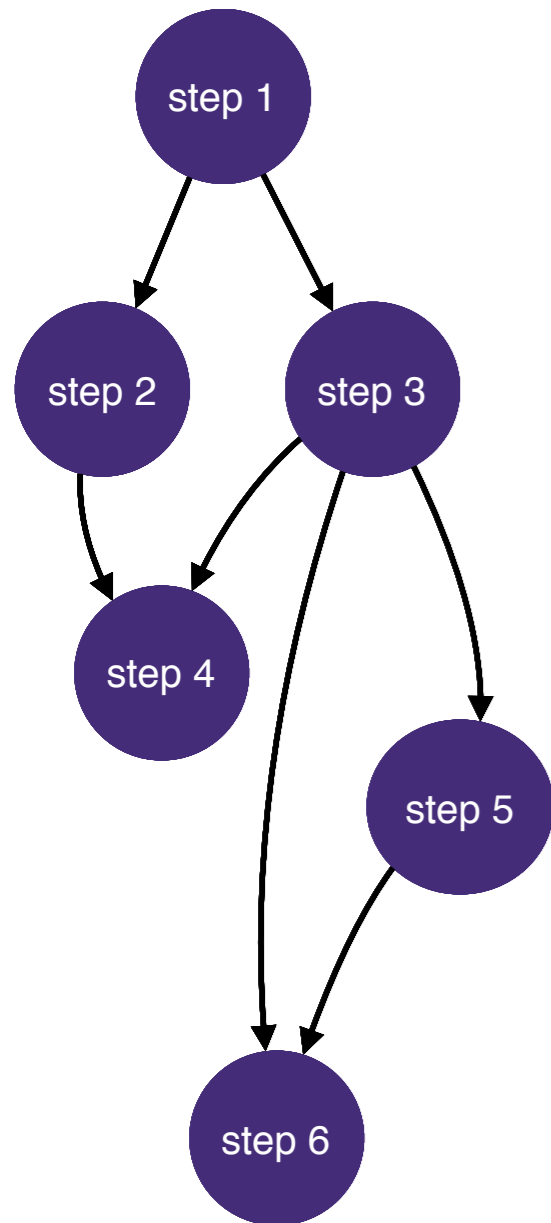
$$f_{\text{analysis}}(\cdot)$$

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**

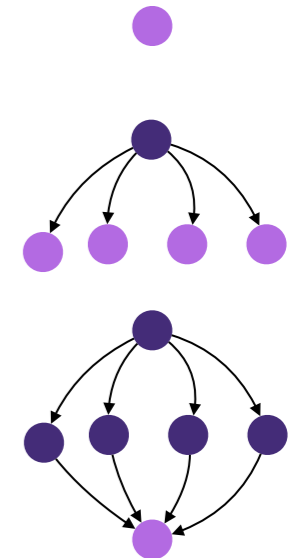


Par. Set 1

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

