Fortgeschrittene Analyse

Christopher Bock, Alexander Mann

Blockkurs: Datenauswertung in der Teilchenphysik
Sommersemester 2015

Einleitung

P1-Praktikum Fluessigkeitsmechanik

- Versuchsaufbau passt auf einen Tisch
- Messwerte werden abgelesen
- Abzaehlbare Anzahl an Messwerten und Observablen
- Auswertung mit Stift und Papier moeglich

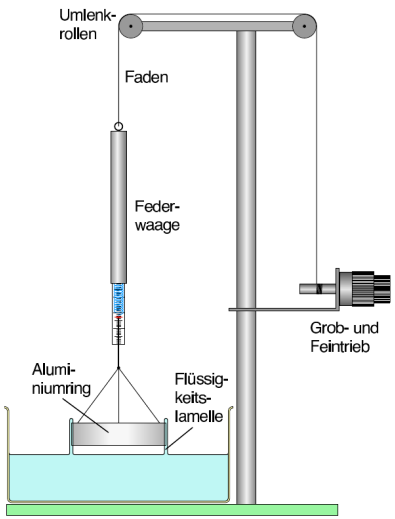
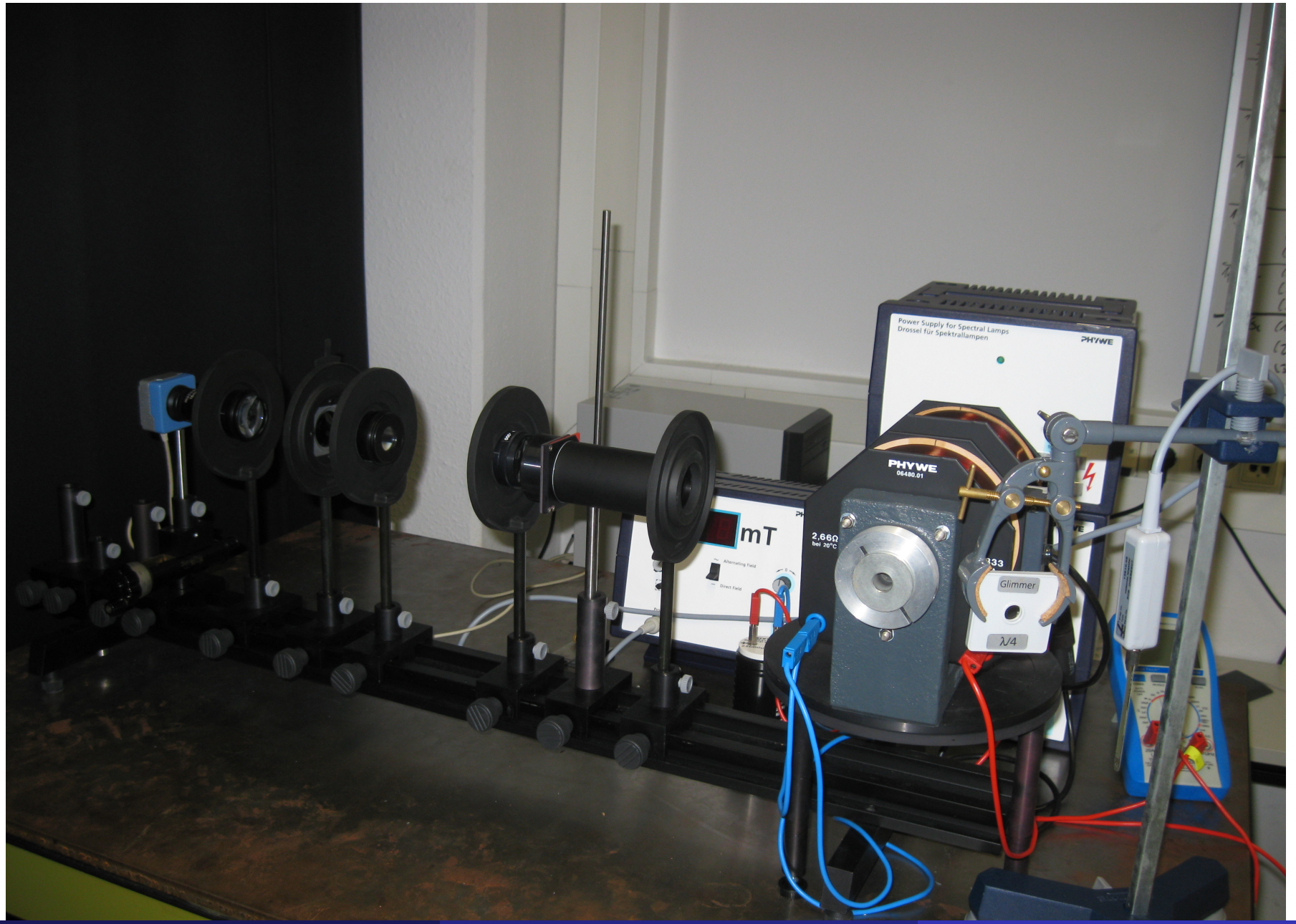
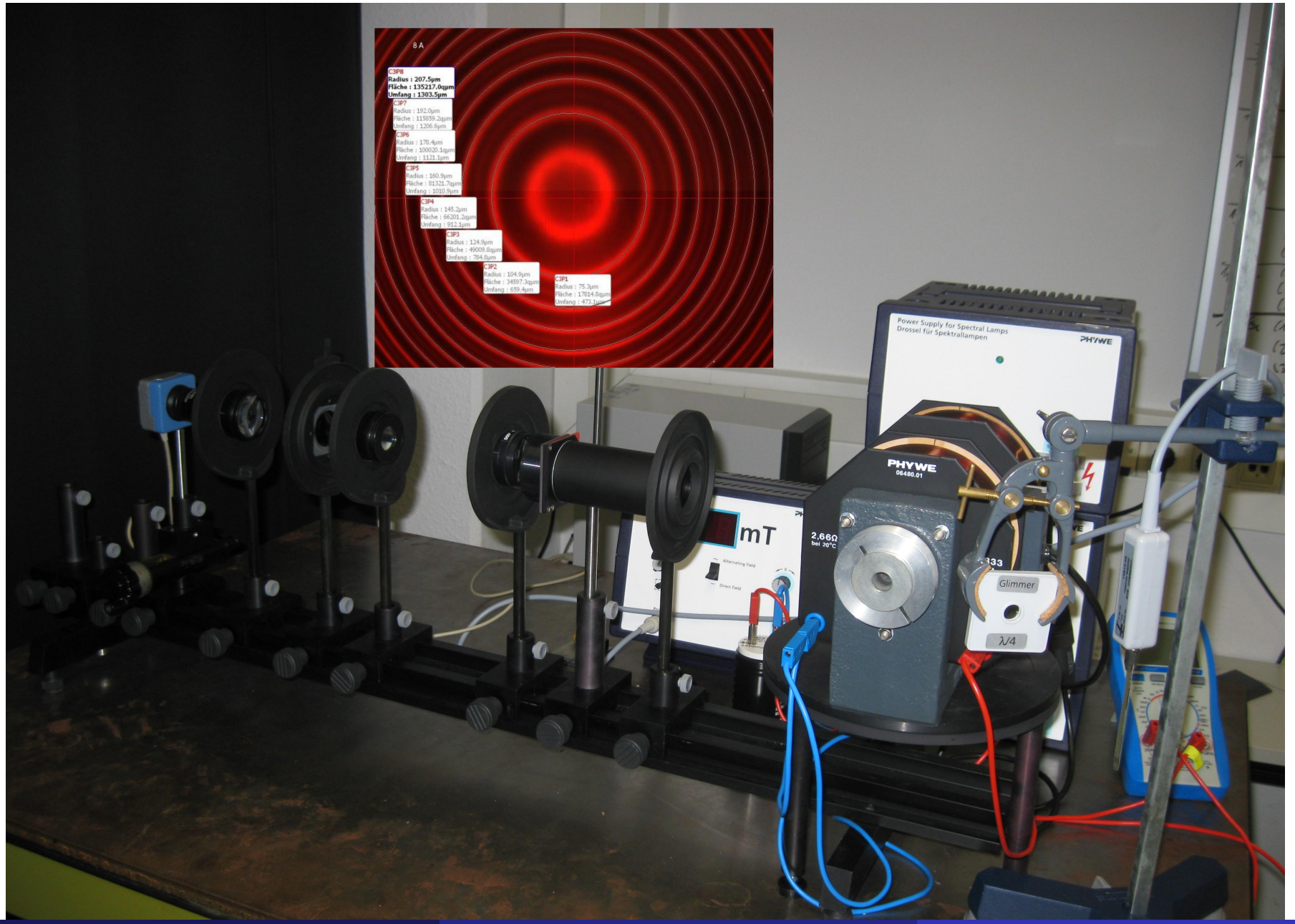


Abbildung 11: Apparatur zur Messung der Oberflächenspannung einer Flüssigkeit nach der Abreißmethode

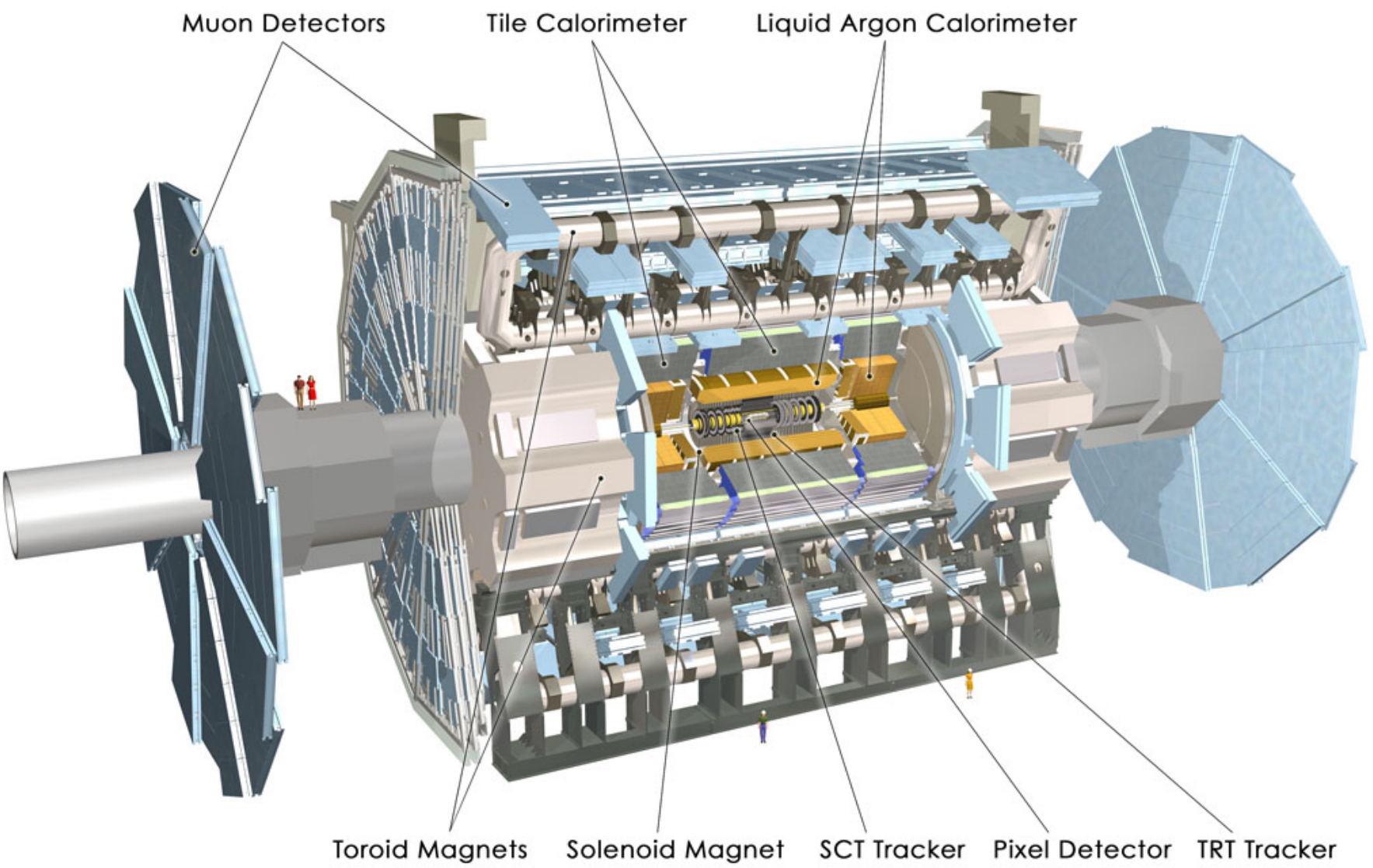
Experimentalphysik



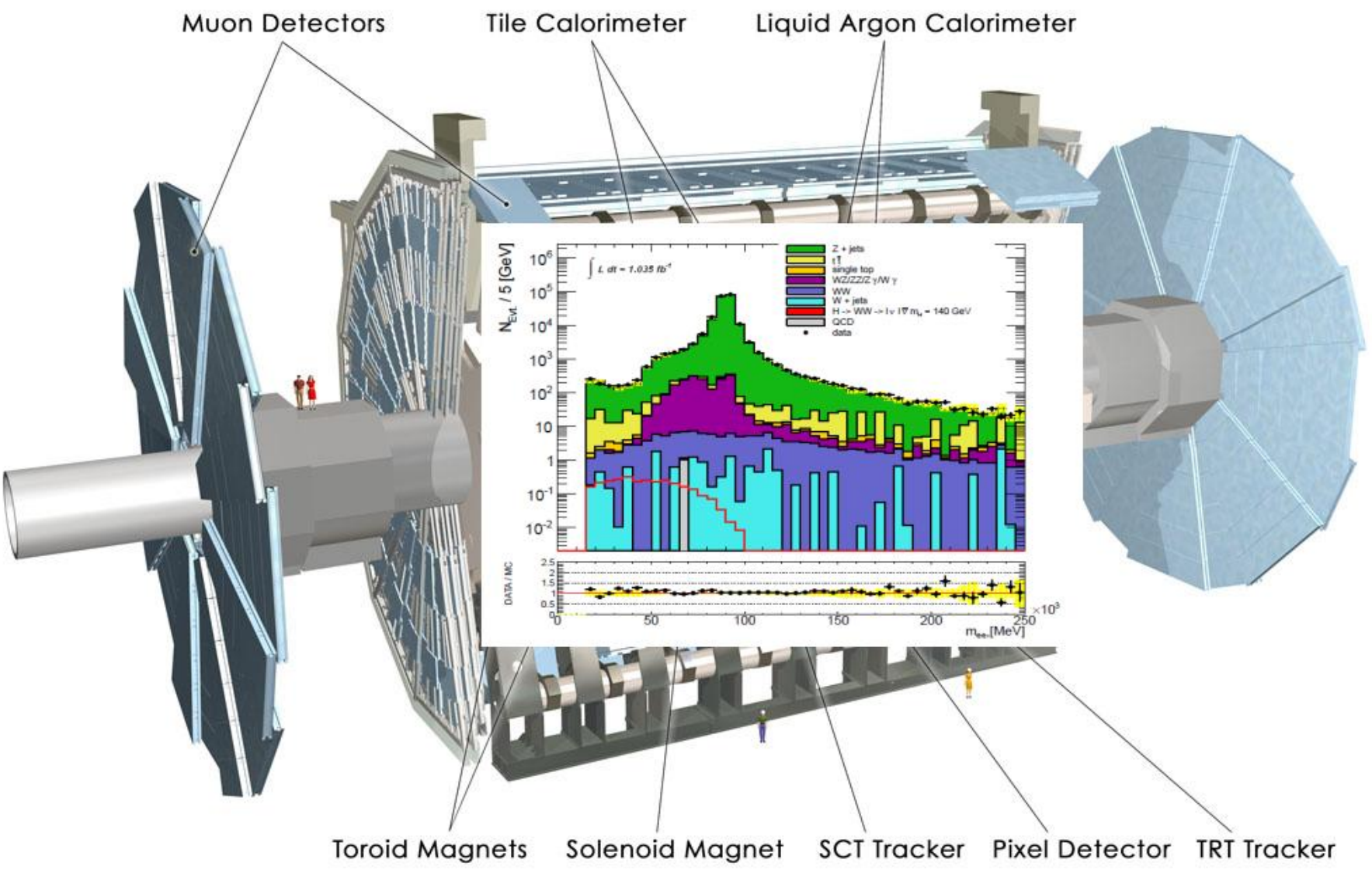
Experimentalphysik



Experimentalphysik



Experimentalphysik



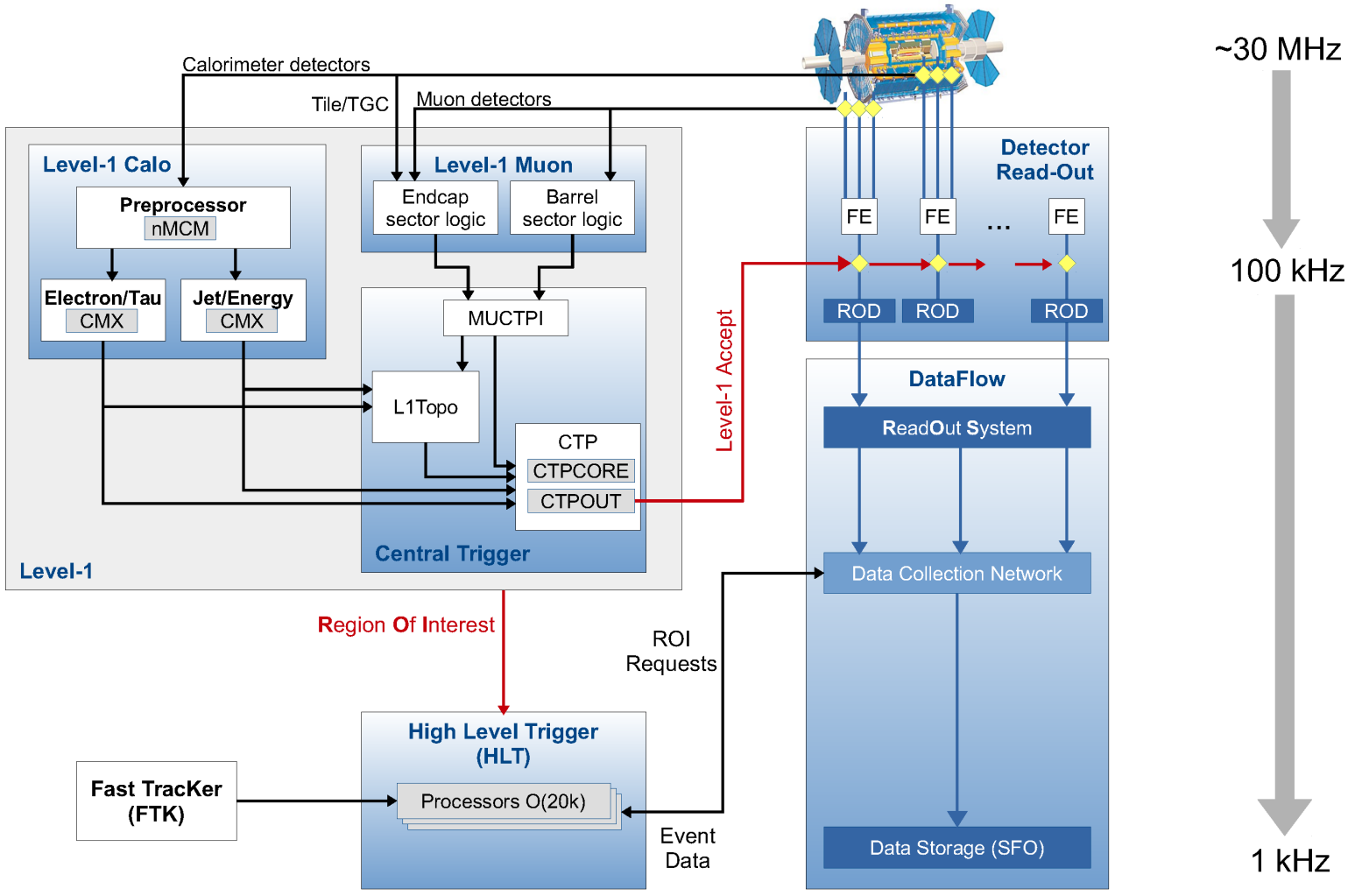
In der Teilchenphysik

- Versuchsaufbauten werden immer komplexer
- Einzelne Messungen sind nicht ausreichend um Schlussfolgerungen zu ziehen
- Viele Observablen
- Verknuepfung vieler Informationen noetig
  - Treffer im Spurdetektor werden zu Spuren verknuepft
  - Energiedepositionen im Kalorimeter werden zu Schauern verknuepft
  - Schauer und Spuren werden zu physikalischen Objekten verknuepft
- Immense Datenmengen entstehen

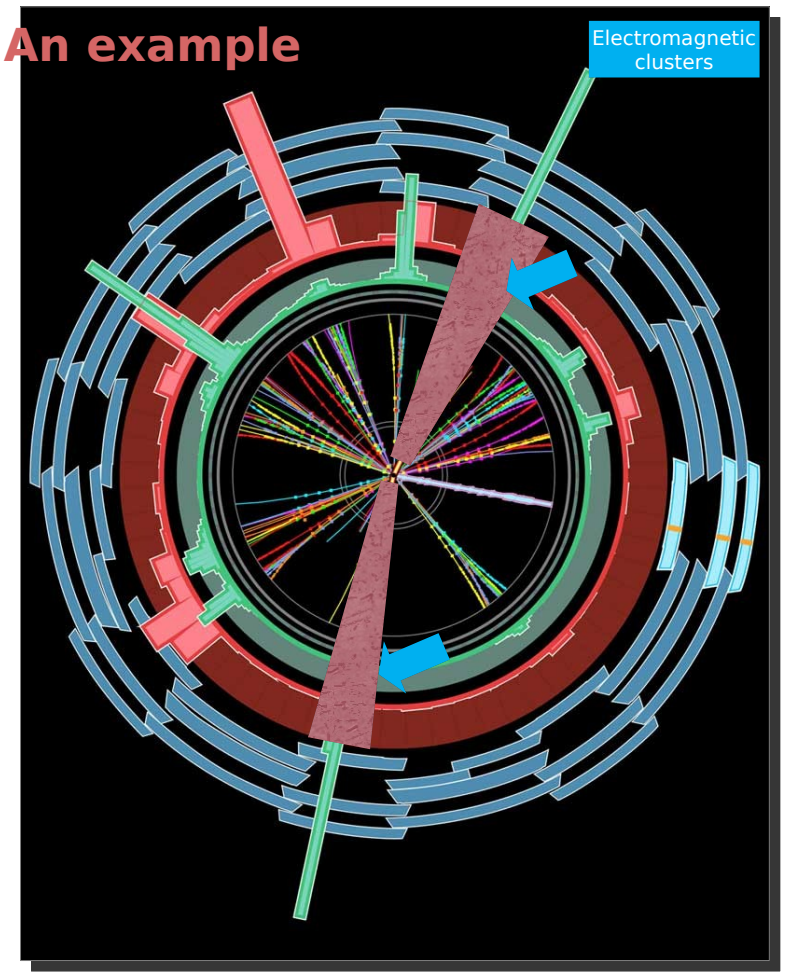
Das Triggersystem

- Aufzeichnung jedes Ereignisses wuerde bedeuten:
  - Bei 50ns bunch spacing
  - 20.000.000 Ereignisse pro Sekunde
  - Annahme: 1MB pro Ereignis
  - Etwa 20 TB an Daten pro Sekunde
  - $\rightarrow$ unmoeglich
  - $\rightarrow$ treffe Vorauswahl $\rightarrow$ Triggersystem

ATLAS Trigger/DAQ system in Run-2



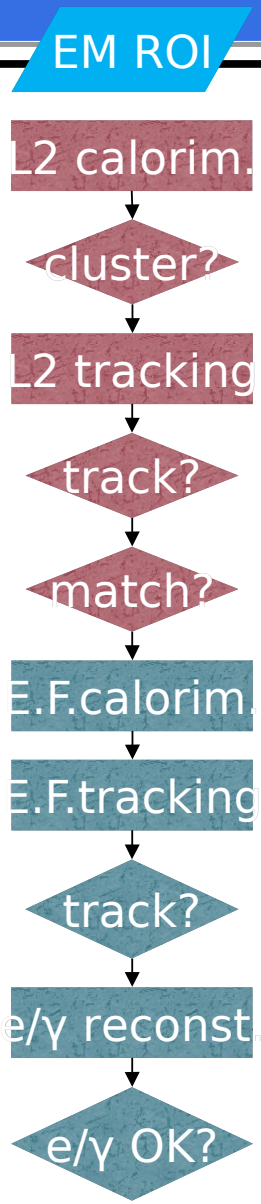
The Trigger



Level1:
Region of Interest is found and position in EM calorimeter is passed to Level 2

Level 2 seeded by Level 1
- Fast reconstruction algorithms
- Reconstruction within RoI

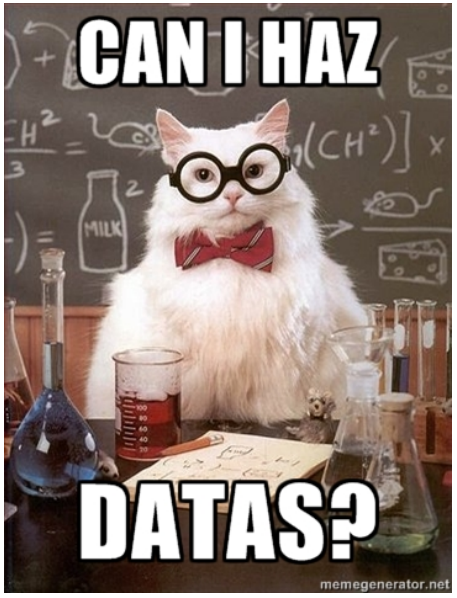
Ev.Filter seeded by Level 2
- Offline reconstruction algorithms
- Refined alignment and calibration



Experimentalphysik

Datenausgabe des LHC: $\sim 15\,\mathrm{PB}$/Jahr $\rightarrow$ “1 DVD alle 10 Sekunden”

Anfang 2013: $\sum \gtrsim 100\,\mathrm{PB}$ (nicht nur LHC-Experimente, aber wesentlicher Anteil vom LHC)

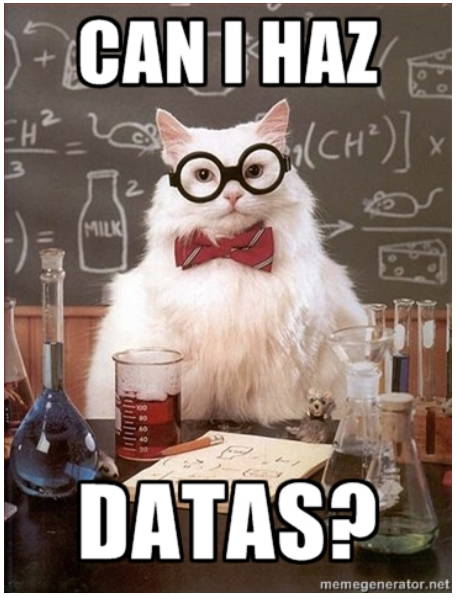


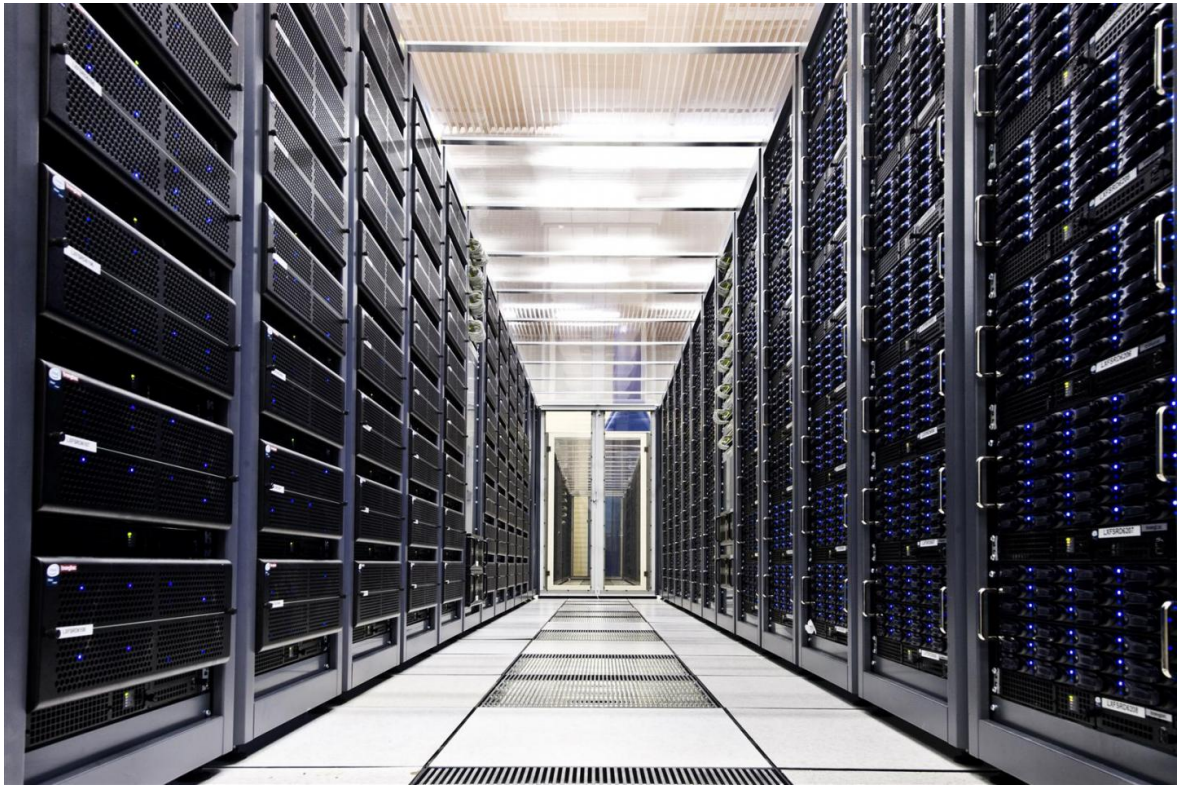
Wie werden wir der Datenflut Herr?

Experimentalphysik

Datenausgabe des LHC: $\sim 15\,\mathrm{PB}$/Jahr $\rightarrow$ “1 DVD alle 10 Sekunden”

Anfang 2013: $\sum \gtrsim 100\,\mathrm{PB}$ (nicht nur LHC-Experimente, aber wesentlicher Anteil vom LHC)



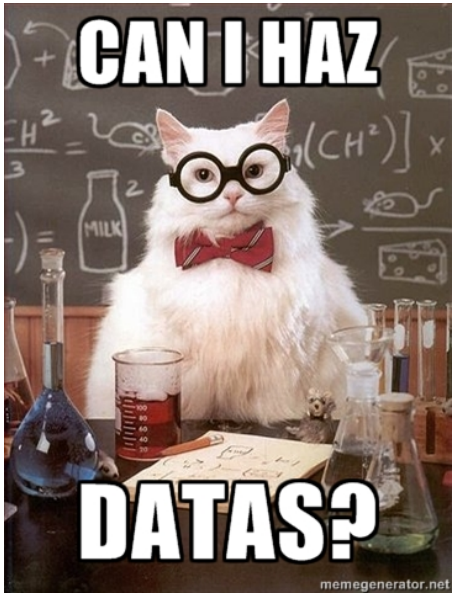


Wie werden wir der Datenflut Herr?

Experimentalphysik

Datenausgabe des LHC: $\sim 15\,\mathrm{PB}$/Jahr $\rightarrow$ “1 DVD alle 10 Sekunden”

Anfang 2013: $\sum \gtrsim 100\,\mathrm{PB}$ (nicht nur LHC-Experimente, aber wesentlicher Anteil vom LHC)

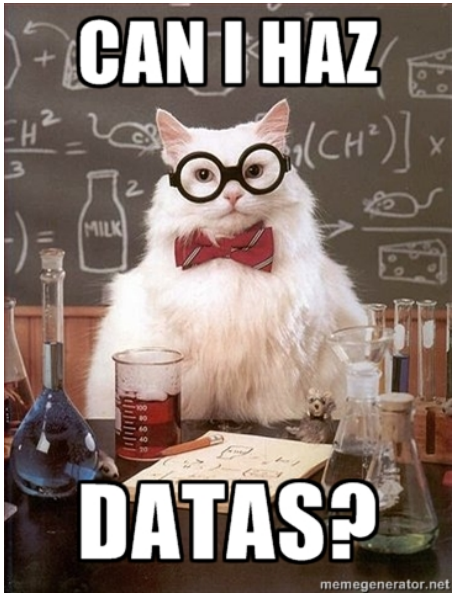


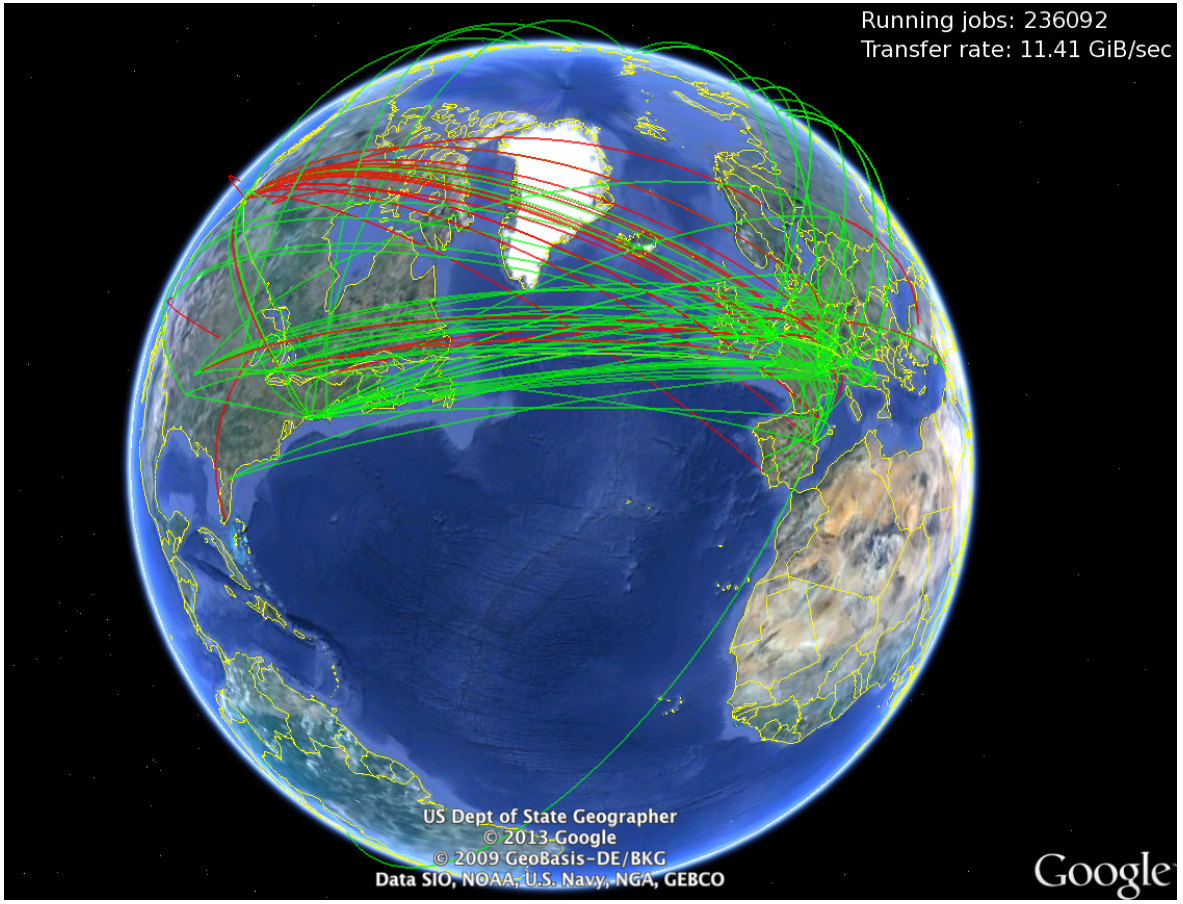
Wie werden wir der Datenflut Herr?

Experimentalphysik

Datenausgabe des LHC: $\sim 15\,\mathrm{PB}$/Jahr $\rightarrow$ “1 DVD alle 10 Sekunden”

Anfang 2013: $\sum \gtrsim 100\,\mathrm{PB}$ (nicht nur LHC-Experimente, aber wesentlicher Anteil vom LHC)





D3PDs

Event Builder

RAW

Tier 0

Reprocessing of full data,
Managed Tape
Access: RAW, ESD, AOD
Disk Access: AOD, parts ESD

RAW/
AOD/ESD

Production of MC events
User Analysis
Disk Storage: AOD

CERN Analysis Facility

1st pass calibrations, Reconstruction, 24h Data export

Tier 1

10 Sites

AOD

Limited Access:
ESD, RAW, calibration
Analysis

+ many Tier3

Tier 2

~40 sites

D3PDs

Datenformate bei ATLAS: NTUP_X vs. (x)AOD

- Rohe Detektordaten (RAW)
  - Gemessene Energiedeposition im Kalorimeter
  - Hits/Treffer im Spurdetektor
  - ...
- $\rightarrow$ Objektrekonstruktion, Verknuepfung aufgezeichneter Informationen zu 'physikalischen' Objekten
- $\rightarrow$ Beziehungen zwischen Objekten
  - z. B. Elektronen und Spuren, Jets und Kalorimeterzellen
  - erlaubt Navigieren zwischen Objekten
  - Repräsentation der Beziehungen als verknüpfte Objekte (C++-Klassen)
  - $\rightarrow$ `(x)AOD`-Format ((extended) Analysis Object Data)
  - `DxAOD`-Format: trifft Vorselektion der benoetigten Informationen (derived xAOD)
  - Analogie: Regal, jedes Fach kann Objekte unterschiedlicher Struktur beherbergen
- 'Frueher': vereinfachte, "flache" Formate
  - Analogie: Excel Tabelle, jeder Eintrag hat die selbe Struktur
  - Verschiedene Variationen: NTUP_SUSY, NTUP_SUSYSKIM, NTUP_QCD, NTUP_JETMET, NTUP_SMWZ, ...
  - + weitere verschlankte Versionen in Untergruppen, "slimmed", "skimmed"

Beispiel

Row	Instance	tau_pt	tau_eta	tau_phi	tau_nPron	el_pt	el_eta	el_phi
0	0	29641.224	1.9894745	1.2342525	2	6562.8969	-1.286600	-2.163984
0	1	26310.502	-1.250536	-2.149304	2	5287.6196	2.0043277	1.1796942
0	2	15408.668	0.8587299	2.1267666	1	5928.1005	-1.216790	-2.178536
0	3	10536.066	-2.245234	0.7030230	1	3567.9082	-0.716857	2.1080973
0	4	4200.5371	1.9659756	1.9340467	1	4185.6108	2.4602475	-1.863633
0	5	12792.959	-0.393817	0.3468983	1	3725.6543	-2.294676	0.7484860
0	6	13422.531	1.1998831	-0.669816	3	2603.2778	-0.879234	-0.147962
0	7	11263.793	1.6567646	-1.989964	1	2576.2331	2.0293889	1.4466400
0	8	9862.3115	1.2783823	-1.282399	0	3180.4751	0.6766212	-0.427249
0	9	14834.677	0.5139963	0.2653455	1	3329.9802	0.9376612	2.4574291
0	10	5577.3891	-2.240545	-0.950781	0	1530.0067	1.4789772	1.5208621
0	11	10008.942	0.0217580	-2.934743	1	3739.2846	0.9350552	2.3634433
0	12	13070.867	-0.988535	-0.363053	0	3252.4023	-2.387143	2.0067243
0	13	12905.367	-0.868744	1.9824419	1	9839.6474	-3.712517	0.1910931
0	14	7583.3867	2.0136349	0.0538617	2	7373.9121	-3.667240	1.8582344
0	15	*	*	*	*	7189.4150	-3.544844	1.6010528
0	16	*	*	*	*	6454.2627	4.2652096	-1.412247
0	17	*	*	*	*	6405.3608	-3.852164	-2.047748
0	18	*	*	*	*	7555.7460	-2.966383	1.4572800
0	19	*	*	*	*	6305.1792	3.5996582	-1.263254
0	20	*	*	*	*	6254.4072	-4.034049	0.3355770
0	21	*	*	*	*	5982.4536	3.6149644	-0.145882
0	22	*	*	*	*	5572.3383	3.5109384	-2.789103
0	23	*	*	*	*	5734.0786	-2.811071	-0.067764
0	24	*	*	*	*	5255.5078	2.7295141	1.8882839

Type <CR> to continue or q to quit ==>

0	25	*	*	*	*	5288.6762	2.8495652	-1.670113
1	0	23161.464	0.9242594	0.4225868	3	8442.6689	0.9072394	0.2988340
1	1	5424.6494	-1.885269	1.8983048	0	5186.5268	-1.647651	2.5627579
1	2	12833.652	0.6049740	-1.236339	1	4205.6699	0.9676566	0.5425505
1	3	9832.9853	-0.894532	-1.521581	1	7262.4233	1.0234866	0.2825859
1	4	7422.0029	-1.737811	-1.511456	0	7485.3896	0.9838590	0.4083353
1	5	*	*	*	*	11106.566	4.2746162	2.5975980
2	0	8615.3232	1.9952153	-2.183061	2	10852.502	-2.274817	0.6940504
2	1	10403.742	2.3160224	1.0473761	0	3315.7043	1.5245029	-1.110899

Typischer Aufbau einer Analyse in der Teilchenphysik

Was brauchen wir?

- Schauen wir doch mal rein. . .
- G. Aad et al. (ATLAS Collaboration):
  "Search for supersymmetry in final states with jets, missing transverse momentum and one isolated lepton in $\sqrt{s} = 7\,\mathrm{TeV}$ pp collisions using $1\,\mathrm{fb}^{-1}$ of ATLAS data"
  Physical Review D 85, 012006 (2012), DOI: 10.1103/PhysRevD.85.012006 *(Empfehlung!)*

PHYSICAL REVIEW D **85,** 012006 (2012)

Search for supersymmetry in final states with jets, missing transverse momentum and one isolated lepton in $\sqrt{s} = 7$ TeV pp collisions using 1 fb^{-1} of ATLAS data

G. Aad *et al.**
(ATLAS Collaboration)
(Received 29 September 2011; published 18 January 2012)

We present an update of a search for supersymmetry in final states containing jets, missing transverse momentum, and one isolated electron or muon, using 1.04 fb^{-1} of proton-proton collision data at $\sqrt{s} = 7$ TeV recorded by the ATLAS experiment at the LHC in the first half of 2011. The analysis is carried out in four distinct signal regions with either three or four jets and variations on the (missing) transverse momentum cuts, resulting in optimized limits for various supersymmetry models. No excess above the standard model background expectation is observed. Limits are set on the visible cross section of new physics within the kinematic requirements of the search. The results are interpreted as limits on the parameters of the minimal supergravity framework, limits on cross sections of simplified models with specific squark and gluino decay modes, and limits on parameters of a model with bilinear R-parity violation.

DOI: 10.1103/PhysRevD.85.012006 PACS numbers: 12.60.Jv, 13.85.Rm, 14.80.Ly

I. INTRODUCTION AND ANALYSIS OVERVIEW

Many extensions of the standard model predict the existence of new colored particles, such as the squarks ($\tilde{q}$) and gluinos ($\tilde{g}$) of supersymmetric (SUSY) theories [1], which could be accessible at the LHC. The dominant SUSY production channels are assumed to be squark-(anti)squark, squark-gluino, and gluino-gluino pair production. Squarks and gluinos are expected to decay to quarks and gluons and the SUSY partners of the gauge

in the first half of 2011. The analysis proceeds similarly to the analysis of the 2010 data [4], with a number of differences. To cover a broader range of signals, the analysis has been extended from one signal search region to four. The kinematic requirements on leptons and jets have been modified, to accommodate changing trigger requirements, minimize the overlap with searches in other final states, and optimize the sensitivity of the search.

As in the 2010 analysis, a combined fit to the observed

Typischer Aufbau einer Analyse in der Teilchenphysik

IV. MONTE CARLO SIMULATION

MC simulations are used to develop the analysis, extrapolate backgrounds from the control to the signal regions, and to assess sensitivity to specific SUSY signal models. Samples of W and Z/γ^* production with accompanying jets are simulated with ALPGEN [33], using the CTEQ6L1 [34] parton density functions (PDFs). Top quark pair production is simulated with MC@NLO [35] and the next-to-leading-order (NLO) PDF set CTEQ66 [36], which is used for all NLO MC. Single top production is simulated with MC@NLO. Fragmentation and hadronization for the ALPGEN and MC@NLO samples is performed with HERWIG [37], using JIMMY [38] for the underlying event. Diboson production is simulated with HERWIG, using the MRST2007LO* [39] modified leading-order PDFs. SUSY signal samples in the MSUGRA/CMSSM model and for the simplified models are generated with HERWIG ++ [40], normalized using NLO cross sections determined by PROSPINO [41]. The bRPV sparticle spectrum is calculated with SPHENO 3.1 [42,43], the event generation is carried out by PYTHIA6 [44] and the NLO cross sections are also provided by PROSPINO. The MC samples are produced using an ATLAS parameter tune of PYTHIA and HERWIG/JIMMY [45] and a GEANT4 [46] based detector simulation [47]. Detailed comparisons of MC-predicted lepton reconstruction and identification efficiencies to the corresponding measurements from data are used to determine scale factors. These scale factors obtained from specifically selected event samples, such as $Z \to \ell\ell$, are then used to correct the MC prediction of efficiencies and acceptances for both signal and background events. The MC samples are produced with a simulation of multiple interactions per LHC bunch crossing (pileup). Differing pileup conditions as a function of the instantaneous luminosity of the LHC machine are taken into account by reweighting MC events according to the mean number of interactions expected.

V. OBJECT RECONSTRUCTION

Collision events are selected by requiring a reconstructed primary vertex with at least five associated tracks, consistent with the beam spot position.

Electrons are reconstructed from clusters in the EM calorimeter matched to a track in the inner detector [48]. Several requirements on the track and clusters are imposed to select true electrons. The "medium" electron selection, used in this analysis to estimate the contribution from nonisolated and misidentified electrons and to veto on dileptonic events, is based on calorimeter shower shape, inner-detector track quality, and track-to-calorimeter-cluster matching. Electrons in the final selection are required to pass the "tight" electron definition, which adds a requirement on the ratio E/p, where E is the calorimeter cluster energy and p is the track momentum, and detection of transition radiation in the TRT. Furthermore, the electron is required to be isolated: the p_T sum of tracks within a cone of $\Delta R < 0.2$ around the electron candidate (excluding the electron candidate itself) is required to be less than 10% of the electron p_T. All electrons are required to pass kinematic cuts of $p_T > 20$ GeV and $|\eta| < 2.47$. In addition, electrons with a distance to the closest jet of $0.2 < \Delta R < 0.4$ are discarded, where $\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2}$. For tight electrons, the p_T requirement is raised to 25 GeV.

Preselected muons are either the result of a combined track in the muon spectrometer and in the inner detector, or a muon spectrometer segment matching with an extrapolated inner detector track [49]. The matched inner detector track must have ≥ 1 hit in the pixel detector, ≥ 1 hit in the inner layer of the pixel detector if the pixel detector module at that location is operational, ≥ 6 hits in the SCT, and fewer than two missing hits on the track in pixel and SCT detectors. For $|\eta| < 1.9$, at least 6 TRT hits are required, and the number of TRT hits that are classified as "outliers" must be less than 90% of the total number of TRT hits on the track. The latter cut is also applied if $|\eta| \geq 1.9$ and at least 6 TRT hits are on the track. TRT outliers appear in two forms in the track reconstruction, as a straw tube with a signal but not crossed by the nearby track, or as a set of TRT measurements in the prolongation of a track which, however, failed to form a smooth trajectory together with the pixel and SCT measurements. These quality cuts are put in place to suppress fake tracks and discriminate against muons from hadron decays. Muons with a distance to the closest jet of $\Delta R < 0.4$ are discarded. In order to reject muons resulting from cosmic rays, tight cuts are applied on the proximity of the muon trajectories to the primary vertex (PV): $|z_\mu - z_{\rm PV}| < 5$ mm and $d_0 < 2$ mm, where z_μ is the z coordinate of the extrapolated muon track at the point of closest approach to the primary vertex, $z_{\rm PV}$ is the z coordinate of the primary vertex, and d_0 is the magnitude of the impact parameter of the muon in the transverse plane. These preselected muons, similar to the electron case, are used to quantify the contribution from nonisolated muons and to reject events with additional muons, and are required to have $p_T > 10$ GeV, and $|\eta| < 2.4$. For muons in the final selection, the p_T requirement is raised to 20 GeV, and the muon is required to be isolated: the p_T sum of tracks within a cone of $\Delta R < 0.2$ around the muon candidate (excluding the muon candidate itself) is required to be less than 1.8 GeV.

Jets are reconstructed using the anti-k_t jet clustering algorithm [50] with a radius parameter of 0.4. The inputs to the jet algorithm are three-dimensional clusters formed from energy deposits in the calorimeter. The jets are calibrated using p_T- and η-dependent correction factors based on MC simulation and validated by test beam and

overlap removal
(zwischen Elektronen und Jets)

Vorauswahl von
Myonkandidaten

(Qualitätskriterien)

overlap removal
(zwischen Myonen und Jets)

Endgültige Myonselektion

Vorauswahl von Jets

b-Jets, fehlende trans-
versale Energie usw.

Selektion von Kollisionsereignissen
anhand der Primärvertizes

Vorauswahl von
Elektronkandidaten

(Qualitätskriterien)

Endgültige Elektronselektion

Zusammenfassung: Prozessieren der Daten

- Grundlegende Datenselektion
  - *Good-runs list (GRL)*
  - Primärvertex
  - Trigger
- Objektvorselektion (*object preselection*)
- Entfernen von Rekonstruktionsdoppeldeutigkeiten (*overlap removal*)

Software-Frameworks (Auswahl)

- EventLoop
- SFrame

3.6 Overlap Removal

To avoid duplication of a particular object in more than one baseline particle collection, the following overlap removal procedure is applied. The overlap removal procedure has been standardized between this analysis and the SUSY electro-weak production analyses with two light leptons, three leptons or four leptons. All steps of the overlap removal are applied consecutively, i.e. for a subsequent step, only baseline particles are considered which survived the overlap removal procedure of the former step. Step 7 specifically removes low mass resonances in the spectrum of the light baseline leptons.

1. $\Delta R(e_1, e_2) \geq 0.05$: If any two baseline electrons (e_1 and e_2) lie within a distance $\Delta R < 0.05$ of each other, the electron with the lower cluster energy E_T is rejected.

2. $\Delta R(e, j) \geq 0.2$: If the distance between a baseline electron (e) and a jet (j) is smaller than 0.2, the jet is rejected.

3. $\Delta R(e/\mu, \tau) \geq 0.2$: If the distance between a baseline electron/muon (e/μ) and a *loose* or baseline tau (τ) is smaller than 0.2, then the tau is rejected.

4. $\Delta R(j, e/\mu) \geq 0.4$: If the distance between a remaining jet (j) and a baseline electron/muon (e/μ) still smaller than 0.4, then the baseline electron/muon is rejected.

5. $\Delta R(e, \mu) \geq 0.01$: If a baseline electron and a baseline muon lie within a distance of $\Delta R < 0.01$, then both the electron and the muon are rejected.

6. $\Delta R(\mu_1, \mu_2) \geq 0.05$: If any two baseline muons (μ_1 and μ_2) lie within a distance of $\Delta R < 0.05$, both muons are rejected.

7. $m(e_1^{\pm}, e_2^{\mp}/\mu_1^{\pm}, \mu_2^{\mp}) \geq 12$ GeV: If the invariant mass of any baseline electron or muon pair with opposite sign ($e_1^{\pm}$ and $e_2^{\mp}/\mu_1^{\pm}$ and $\mu_2^{\mp}$) is less than 12 GeV, then both electrons/muons are rejected.

8. $\Delta R(\tau, j) \geq 0.2$: If the distance between a baseline (*medium*) tau (τ) and a jet (j) is smaller than 0.2, the jet is rejected.

Beispiel *overlap removal* in einer 2-Tau-Analyse

Zusammenfassung: Prozessieren der Daten

- Grundlegende Datenselektion
  - *Good-runs list (GRL)*
  - Primärvertex
  - Trigger
- Objektvorselektion (*object preselection*)
- Entfernen von Rekonstruktionsdoppeldeutigkeiten (*overlap removal*)
- Objektselektion (*object selection*)
- Berechnung kinematischer Ereignisgrößen
  (z. B. “effektive Masse” $\sim \sum$ alle Objekte)
- Ereignisselektion (*event selection*)
  - Endzustand
  - weitere Schnitte…
  - $\Rightarrow$ *cutflow*
- Füllen von Histogrammen

Software-Frameworks (Auswahl)

- EventLoop
- SFrame

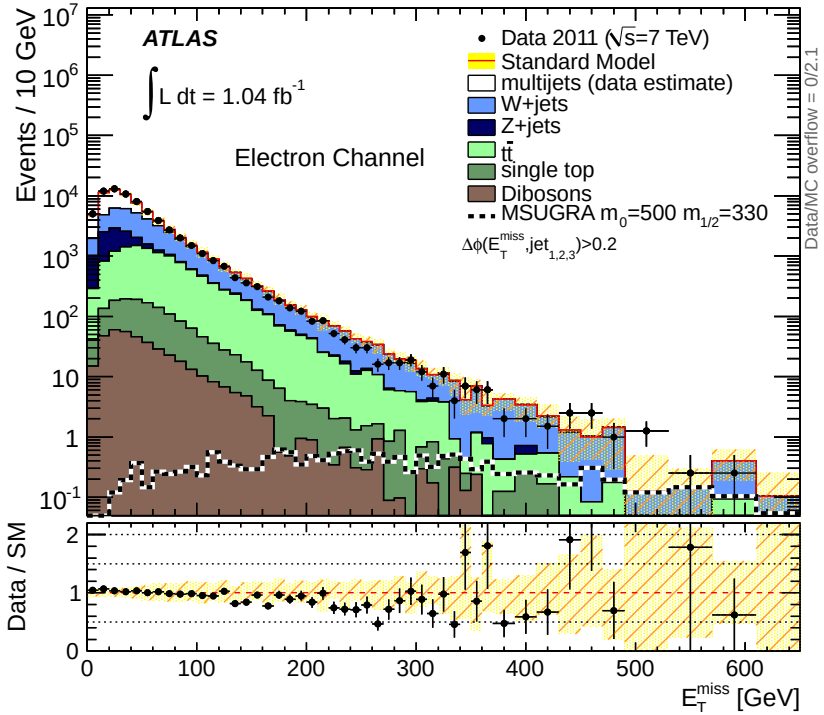
Zusammenfassung: Prozessieren der Daten

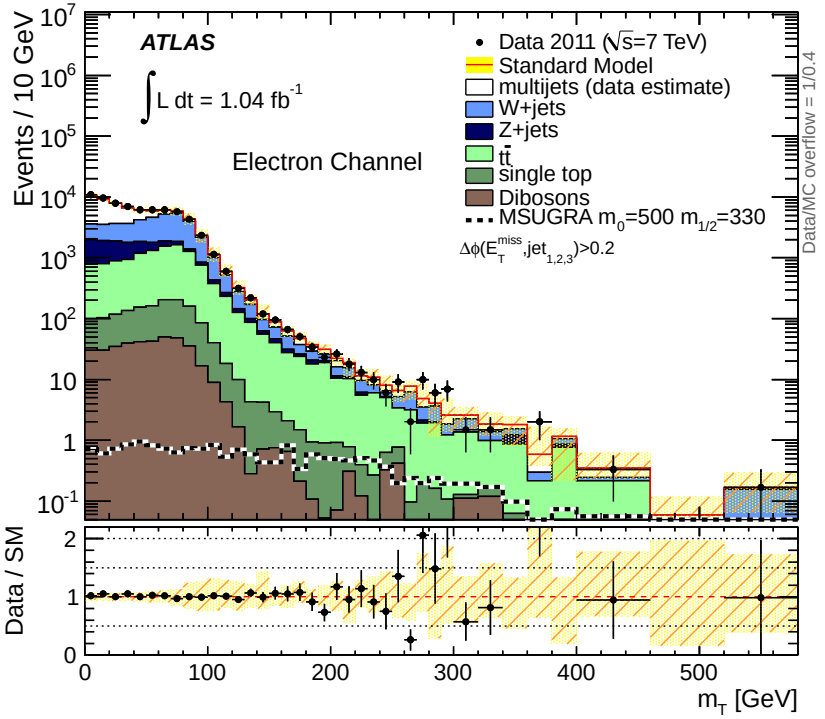
- Grundlegende Datenselektion
  - *Good-runs list (GRL)*
  - Primärvertex
  - Trigger
- Objektvorselektion (*object preselection*)
- Entfernen von Rekonstruktionsdoppeldeutigkeiten (*overlap removal*)
- Objektselektion (*object selection*)
- Berechnung kinematischer Ereignisgrößen
  (z. B. “effektive Masse” $\sim \sum$ alle Objekte)
- Ereignisselektion (*event selection*)
  - Endzustand
  - weitere Schnitte. . .
  - $\Rightarrow$ *cutflow*
- Füllen von Histogrammen

Software-Frameworks (Auswahl)

- EventLoop
- SFrame

Ein paar Resultate





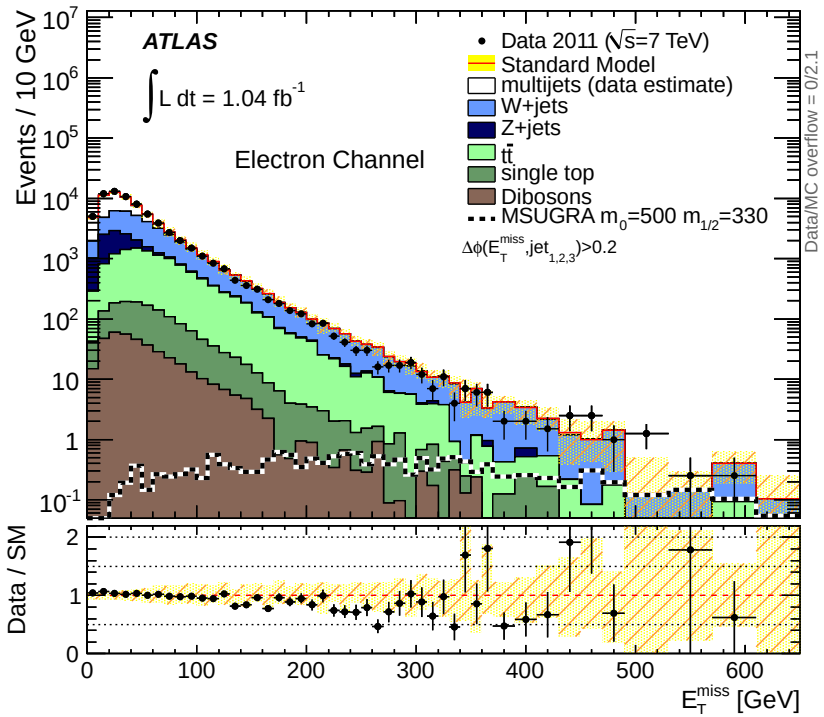
Schnitte

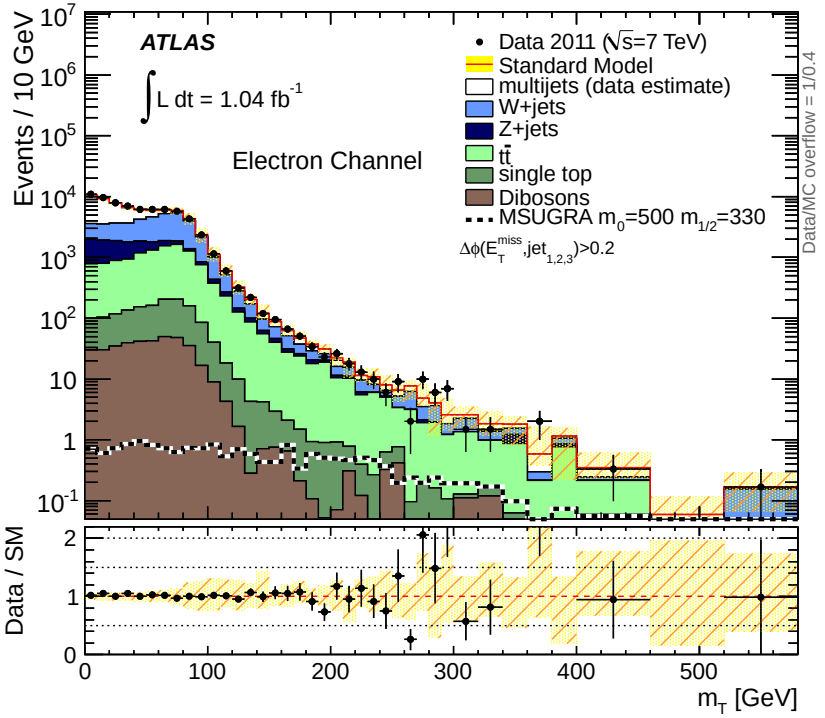
- 1 Elektron mit $p_T > 25$GeV
- Min. 3 Jets mit $p_T > 60, 25, 25$GeV
- $\Delta\phi(\text{jet}_i, \vec{E}_T^{miss}) > 0.2$

Was ist gezeigt?

- **Links:** Fehlende transversale Energie
- **Rechts:** Transverse Masse
- Zusammensetzung des Untergrunds
- Ein Signalpunkt zum Vergleich
- Verhaeltnis von Daten zur SM Vorhersage

Ein paar Resultate





Was lernen wir?

- SM Vorhersage unterschaetzt leicht die fehlende transversale Energie
- Innerhalb der Unsicherheiten (gelbes Band) stimmen Daten und Vorhersage gut ueberein
- Gelbes Band: statistische und systematische Unsicherheiten
- Fehlerbalken der Daten: nur statistische Unsicherheiten

Ein paar Resultate

Electron channel	3JL	3JT	4JL	4JT
Total statistical ($\sqrt{N_{\text{obs}}}$)	± 8.4	± 3.7	± 6.4	± 3.0
Total background systematic	± 30.2	± 7.4	± 17.9	± 3.7
Jet/$E_{\text{T}}^{\text{miss}}$ energy resolution	± 5.9	± 0.5	± 4.2	± 0.8
Jet/$E_{\text{T}}^{\text{miss}}$ energy scale	± 18.6	± 4.1	± 13.6	± 2.4
Lepton energy resolution	± 0.5	± 0.3	± 0.1	± 0.3
Lepton energy scale	± 1.1	± 0.3	± 0.4	± 0.5
b-tagging	± 1.2	± 0.2	± 0.7	± 0.1
MC stat. top	± 5.8	± 2.0	± 3.8	± 1.4
MC stat. W	± 4.4	± 2.3	± 2.2	± 1.3
Lepton misidentification rate	± 1.4	± 0.1	± 0.2	< 0.1
Real lepton rate	± 1.5	± 0.3	± 0.8	± 0.1
Top background modeling	± 15.9	± 2.1	± 9.8	± 1.2
W background modeling	± 19.0	± 5.6	± 5.1	± 1.9
Pile-up	± 5.1	± 1.0	± 2.5	± 0.4

Was ist gezeigt?

Dominante systematische Unsicherheiten gemessen in Anzahl von Ereignissen fuer die verschiedenen Signal Regionen. Nicht alle sind zwangsweise unabhaengig!

Ein paar Resultate

Electron channel	3JL	3JT	4JL	4JT
Total statistical ($\sqrt{N_{\text{obs}}}$)	± 8.4	± 3.7	± 6.4	± 3.0
Total background systematic	± 30.2	± 7.4	± 17.9	± 3.7
Jet/$E_{\text{T}}^{\text{miss}}$ energy resolution	± 5.9	± 0.5	± 4.2	± 0.8
Jet/$E_{\text{T}}^{\text{miss}}$ energy scale	± 18.6	± 4.1	± 13.6	± 2.4
Lepton energy resolution	± 0.5	± 0.3	± 0.1	± 0.3
Lepton energy scale	± 1.1	± 0.3	± 0.4	± 0.5
b-tagging	± 1.2	± 0.2	± 0.7	± 0.1
MC stat. top	± 5.8	± 2.0	± 3.8	± 1.4
MC stat. W	± 4.4	± 2.3	± 2.2	± 1.3
Lepton misidentification rate	± 1.4	± 0.1	± 0.2	< 0.1
Real lepton rate	± 1.5	± 0.3	± 0.8	± 0.1
Top background modeling	± 15.9	± 2.1	± 9.8	± 1.2
W background modeling	± 19.0	± 5.6	± 5.1	± 1.9
Pile-up	± 5.1	± 1.0	± 2.5	± 0.4

Was lernen wir?

- Systematische Unsicherheiten sind hier groesser als die statistischen
- Die systematischen Unsicherheiten werden von der JES und der Modellierung des Top und des W Untergrunds dominiert

Ein paar Resultate

Electron channel	3JL Signal region	3JT Signal region	Top region	W region
Observed events	71	14	162	565
Fitted top events	56 ± 20 (51)	7.6 ± 3.0 (6.8)	125 ± 16 (112)	64 ± 8 (58)
Fitted W/Z events	35 ± 20 (34)	10.5 ± 6.5 (10.1)	30.1 ± 9.1 (29.3)	425 ± 36 (413)
Fitted multijet events	$6.0^{+2.3}_{-1.4}$	$0.46^{+0.37}_{-0.22}$	7.2 ± 2.6	76 ± 24
Fitted sum of background events	97 ± 30	18.5 ± 7.4	162 ± 13	565 ± 24

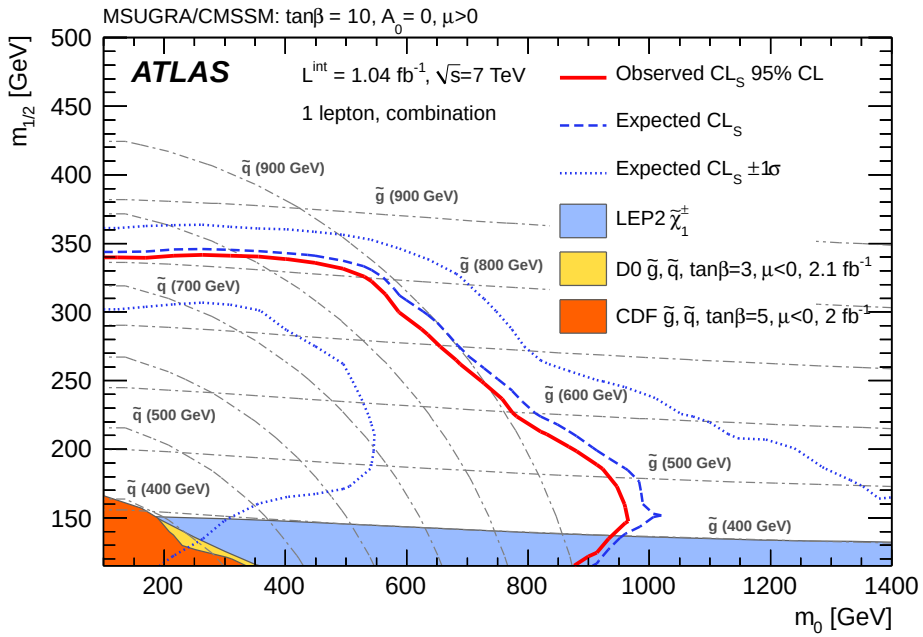
Was ist gezeigt?

Anzahl erwarteter Ereignisse in den Signal- und Validierungsregionen.

Was lernen wir?

- Top und W/Z Untergruende sind dominant
- Koennen die zuvor gezeigten Unsicherheiten in Bezug setzen (Beispiel 3JL)
  - Gesamte syst. Unsicherheit: ± 30.2
  - Modellierung des Top Untergrunds: ± 15.9

Ein paar Resultate



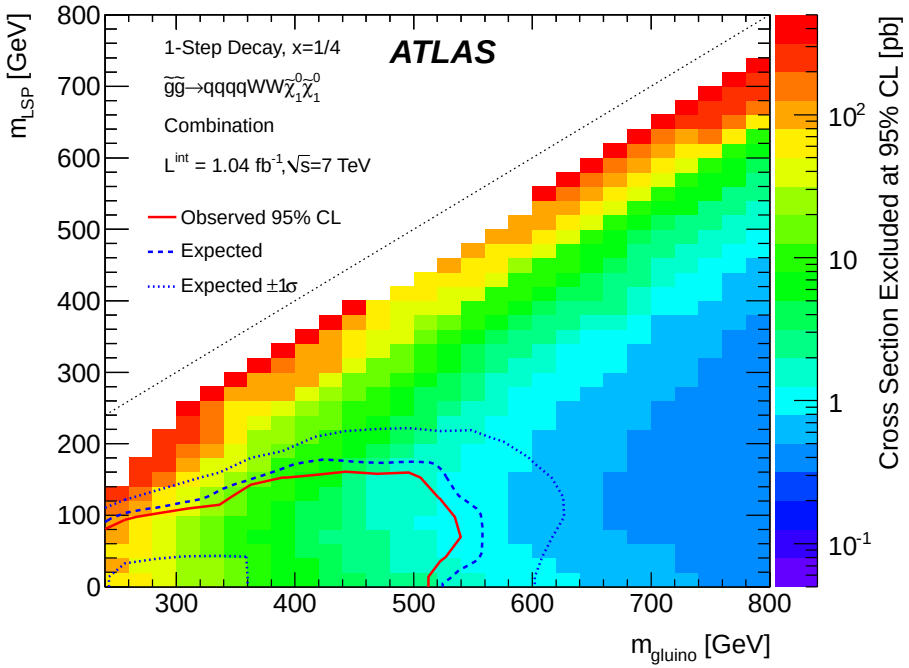
Was ist gezeigt?

Ausschlusskontur im m_0 - $m_{1/2}$ Parameter Raum fuer ein MSUGRA/CMSSM Model mit $\tan(\beta) = 10, \mathrm{A}_0 = 0, \mu > 0$

Was lernen wir?

- Die Kontur hat sich verbessert gegenueber den vorherigen Analysen
- Diese Resultate haben a priori **keine Allgemeingueltigkeit**
- Ausschlussgrenzen **haengen oft stark vom Model** ab in dem sie interpretiert werden

Ein paar Resultate



Was ist gezeigt?

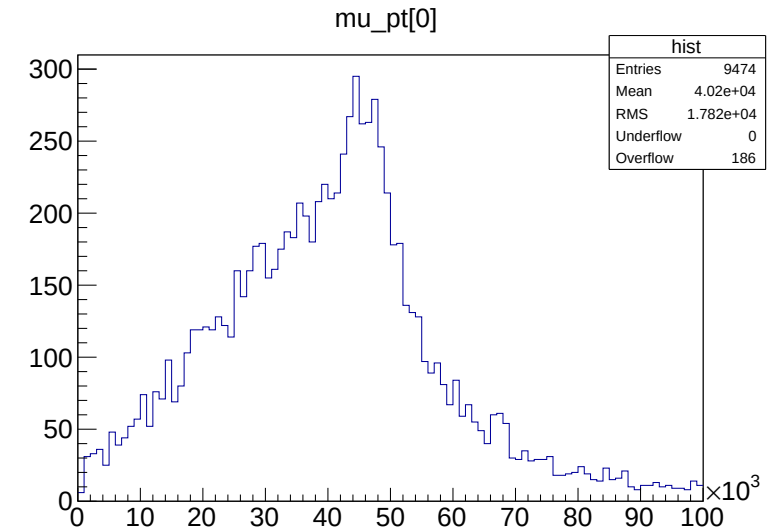
Ausgeschlossener Wirkungsquerschnitt im physikalischen m_{LSP} - m_{gluino} Parameter Raum in einem vereinfachten Model fuer einen spezifischen Prozess

Was lernen wir?

- Welche Wirkungsquerschnitte fuer diesen Prozess 'noch' erlaubt sind
- Wichtiger Input fuer Theoretiker
- Gilt oftmals nur unter gewissen Annahmen

Analyse mit ROOT

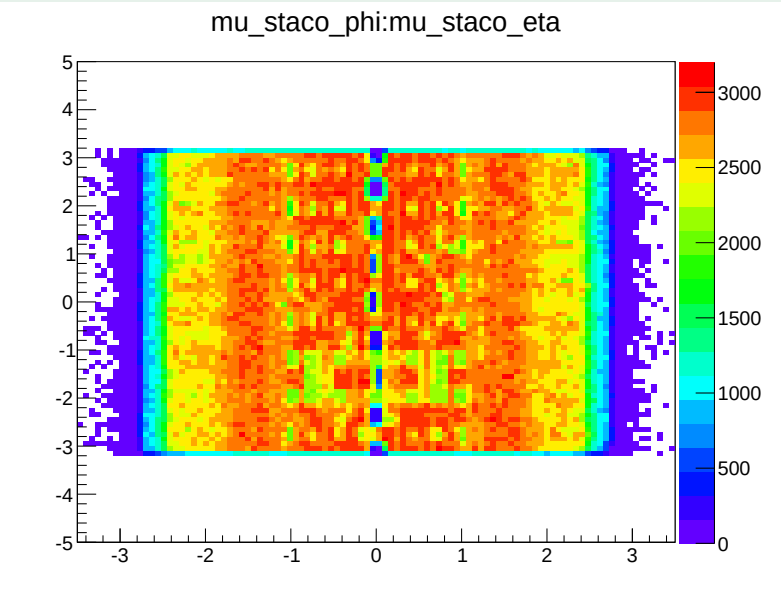
- **Beschränkung auf D3PD / NTUP_X als Dateiformat**
- 1) einfachste ROOT-Analyse
  - `TBrowser` zum Navigieren durch den Inhalt eines D3PDs
  - `TTree::Draw()` zum Plotten von Variablen
  - beschränkt auf sehr einfache Plots und Selektionen




```
tree->Draw("mu_pt[0]");
```

Analyse mit ROOT

- **Beschränkung auf D3PD / NTUP_X als Dateiformat**
- 1) einfachste ROOT-Analyse
  - `TBrowser` zum Navigieren durch den Inhalt eines D3PDs
  - `TTree::Draw()` zum Plotten von Variablen
  - beschränkt auf sehr einfache Plots und Selektionen




```
tree->Draw("mu_staco_phi:mu_staco_eta
>>hist(100,-3.5,3.5, 100, -5,5)", "", "COLZ")
```

Analyse mit ROOT

- **Beschränkung auf D3PD / NTUP_X als Dateiformat**
- 2) komplexere Selektionen mit ROOT:
  - `TTree::MakeSelector()` erzeugt Grundgerüst
  - basierend auf Variablen, die in einem D3PD gefunden werden
  - pro Ereignis kann auf alle Objekte zugegriffen werden, erlaubt auch komplexe Operationen
  - ändert sich die Struktur des Datensatzes, muss Grundgerüst angepasst werden

```
//////////////////////////////////////////////////////////
// This class has been automatically generated on
// Fri Apr 12 14:29:45 2013 by ROOT version 5.34/05
// from TTree h5000/EVENT
// found on file: ntz0e1.root
//////////////////////////////////////////////////////////

#ifndef h5000_h
#define h5000_h

#include <TROOT.h>
#include <TChain.h>
#include <TFile.h>
#include <TSelector.h>

// Header file for the classes stored in the TTree if any.

// Fixed size dimensions of array or collections stored in the TTree if any.

class h5000 : public TSelector {
public :
   TTree          *fChain;   //!pointer to the analyzed TTree or TChain

   // Declaration of leaf types
   Float_t         Run;
   Float_t         Event;
   Float_t         Ncharged;
   Float_t         Pcharged;
   Float_t         N_ecal;
   Float_t         E_ecal;
   Float_t         E_hcal;
   Float_t         Nmuopen;
```

```
tree->MakeSelector();
```

Uebungsaufgabe

Analyse mit ROOT

- Uebungen unter:
  http://www.etp.physik.uni-muenchen.de/kurs/comp15/uebungen/node34.html
- Bis einschließlich 'Eine Selektion starten'

```
//////////////////////////////////////////////////////////
// This class has been automatically generated on
// Fri Apr 12 14:29:45 2013 by ROOT version 5.34/05
// from TTree h5000/EVENT
// found on file: ntz0e1.root
//////////////////////////////////////////////////////////

#ifndef h5000_h
#define h5000_h

#include <TROOT.h>
#include <TChain.h>
#include <TFile.h>
#include <TSelector.h>

// Header file for the classes stored in the TTree if any.

// Fixed size dimensions of array or collections stored in the TTree if any.

class h5000 : public TSelector {
public :
   TTree          *fChain;   //!pointer to the analyzed TTree or TChain

   // Declaration of leaf types
   Float_t         Run;
   Float_t         Event;
   Float_t         Ncharged;
   Float_t         Pcharged;
   Float_t         N_ecal;
   Float_t         E_ecal;
   Float_t         E_hcal;
   Float_t         Nmuonen;
```

```
tree->MakeSelector();
```

SFrame

Vektoren *(zur Erinnerung)*

`std::vector`

- In der Informatik: Vektor = eindimensionales Array
  (in der Regel dem Array mit manueller Speicherverwaltung vorzuziehen)
- Teil der Standard-Template-Bibliothek (STL) von `C++`
- Wichtige Methoden: `begin()`, `end()`, `push_back(...)`, `at(...)`, `clear()`, ...
- Dokumentation (eine von vielen)

`TLorentzVector`

- `C++`-Klasse, beschreibt "relativistische" Vierervektoren im Minkowski-Raum
  - d. h. Vektoren, die sich gemäß der Lorentztransformationsvorschrift transformieren
  - wichtige Beispiele: (ct, x_1, x_2, x_3), $(E/c, p_1, p_2, p_3)$
- Kommt mit ROOT
- Erlaubt einfaches Rechnen mit Vierervektoren
- Wichtige Methoden: `SetXYZT(...)`, `SetPtEtaPhiE(...)`, `Boost(...)`, `DeltaR(...)`, `M()`, `operator+(...)`, ...
- Dokumentation

SFRAME: Übersicht

Vorteile

- Dynamisches Lesen von Variablen
  (Flexibilität: nur tatsächlich verwendete Variablen müssen existieren)
- Konfiguration der Jobs mittels Textdateien (XML)
  (erlaubt schnelle Änderungen an der Konfiguration ohne Neukompilieren)
- Optimiert auf hohen Datendurchsatz, läuft ohne Änderung mit PROOF / PoD
- Paketstruktur erlaubt sinnvolle Organisation des Codes
- Gut dokumentiert, wird aktiv weiterentwickelt, E-Mail-Support durch die Entwickler

- Code / Download: `http://sourceforge.net/projects/sframe/`
  - Link “Download newest release” ist veraltet
  - besser: `svn checkout svn://svn.code.sf.net/p/sframe/code/SFrame/tags/SFrame-03-06-28 SFrame`
- Wiki mit Dokumentation: `http://sourceforge.net/apps/mediawiki/sframe/`
- Doxygen-Dokumentation: `http://sframe.sourceforge.net/Doxygen/`

Einfacher Beispielcode

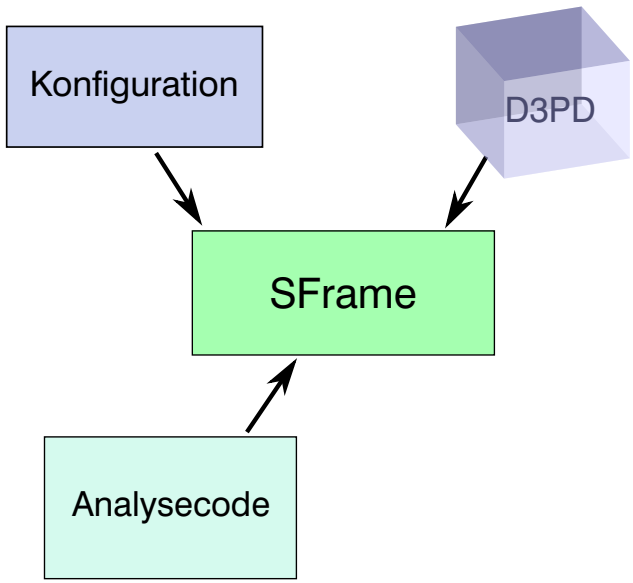
- Archiv mit Code in ein lokales Arbeitsverzeichnis kopieren und entpacken:
  - `/project/etp/Bachelor15/SFrame_First_Steps.tgz`
  - (Kopieren: `cp [von] [nach]`)
  - (Entpacken: `tar zxf [Datei]`)
- In Verzeichnis `work0` wechseln und Setup starten:
  `source sf.sh`
- In Verzeichnis `SFrame/` wechseln und SFRAME kompilieren:
  `make`
- In Unterverzeichnis `user/` wechseln und die Beispielanalyse kompilieren:
  `make`
- In Unterverzeichnis `config/` wechseln und Beispielanalyse laufenlassen mit:
  `sframe_main FirstCycle_config.xml`
- Ausgabe betrachten:
  `root -l FirstCycle.MC.Zee_1.root`
- Was enthält die Ausgabe?
- Was steht in der `FirstCycle_config.xml`?
  - (Hinweis: alles zwischen `<!--` und `-->` sind Kommentare zur Erläuterung)

SFRAME: Blick in den Code

- `FirstCycle()`: Konstruktor
  - verwende `DeclareProperty()`, um Konfigurationsparameter zu definieren
- `BeginInputData()`: wird beim Öffnen eines Datensatzes aufgerufen
  - verwende `DeclareVariable()`, um Ausgabevariablen zu definieren
  - verwende `Book()`, um Ausgabehistogramme zu definieren
- `BeginInputFile()`: wird beim Öffnen einer Eingabedatei aufgerufen
  - verwende `ConnectVariable()`, um Eingabevariablen anzubinden (vgl. `SetBranchAddress()` in ROOT)
- `EndInputData()`: wird nach dem Prozessieren eines Datensatzes aufgerufen
  - z. B. Ausgabe von Statistiken
- `ExecuteEvent()`: wird einmal pro Ereignis aus dem Eingabedatensatz aufgerufen
  - hier findet die eigentliche Arbeit statt!

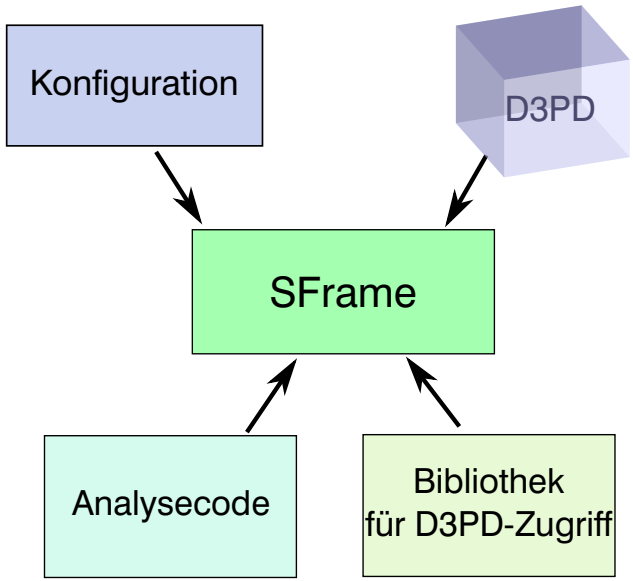
Unterschiede zum EVENTLOOP und anderen

- Umfangreicher und mit mehr Möglichkeiten
- Kompilierter Code (d. h. nicht direkt über ROOT startbar)
- Konfiguration über Textdateien
- Hier haben wir erstmal wirklich nur das Grundgerüst kennengelernt. . .
  - keine Objektselektion
  - kein *overlap removal*
  - . . . und müssen wir wirklich wieder alle Variablen mühsam am Anfang von Hand verbinden?



Unterschiede zum EVENTLOOP und anderen

- Umfangreicher und mit mehr Möglichkeiten
- Kompilierter Code (d. h. nicht direkt über ROOT startbar)
- Konfiguration über Textdateien
- Hier haben wir erstmal wirklich nur das Grundgerüst kennengelernt. . .
  - keine Objektselektion
  - kein *overlap removal*
  - . . . und müssen wir wirklich wieder alle Variablen mühsam am Anfang von Hand verbinden?
    $\rightarrow$ Nein: verwende Bibliothek *(später)*



Vorgehensweise

- Entpacken Sie `/project/etp/Bachelor15/SFrame_MET.tgz`
- Rufen Sie das Setup auf mit `source sf.sh` im entpackten Verzeichnis `SFrame_MET/`.
- Kompilieren Sie den Analysecode im Unterordner `MET` mit `make`.
- Starten Sie die Analyse mit `sframe_main metfile_config.xml` im Unterordner `config/`.
- Betrachten Sie die Log-Ausgabe, öffnen Sie die erzeugte `.root`-Datei und lesen Sie den Analysecode.
  - Finden Sie die mit `DeclareProperty` deklarierten Optionen in der Konfiguration wieder?
  - Wofür dient `ConnectVariable`? Wofür `DeclareVariable`?
  - Wie werden Histogramme definiert und gefüllt?

Binärkonflikte?

- Analysecode verwendet externe Pakete (siehe Ordner`External/`)
- Vorab kompiliert $\rightarrow$ für bestimmte Prozessorarchitektur
- Andere Prozessor (z.B. 32-Bit- statt 64-Bit-Architektur)
  $\rightarrow$ Neukompilieren der Bibliotheken notwendig
- Große Herausforderung bei heterogenen Computerverbünden

Fragen und Aufgaben

- Beheben Sie den in der Ausgabe rot hervorgehobenen Fehler.
- Aktivieren Sie das Rausschreiben eines `TTree`s
  - Was enthält dieser? Dateigröße?
  - Danach wieder deaktivieren...
- Laufen Sie über mehr als 1000 Ereignisse
- Verwenden Sie als Eingabedatensatz die Ereignisse aus dem Minimum-Bias-Stream
  - Zu finden in einem Unterordner von `/project/etp/Bachelor15/data`
  - Sie benötigen dafür einen anderen Trigger, den Sie über die Option “Trigger1” einstellen können: `EF_rd0_filled_NoAlg`
  - Wie sehen die Histogramme jetzt aus?

Einfache Analyse

- Neue “saubere” Konsole öffnen (!)
- Archiv mit Code in ein lokales Arbeitsverzeichnis kopieren und entpacken:
  - Archiv: `/project/etp/Bachelor15/SFrame_D3PD_Analyse.tgz`
  - (Kopieren: `cp [von] [nach]`)
  - (Entpacken: `tar zxf [Datei]`)
- In Verzeichnis `SFrame_D3PD_Analyse1/` wechseln und Setup starten:
  `source sf.sh`
- In Unterverzeichnis `Analyse1/` wechseln und die Analyse kompilieren:
  `make`
- In Unterverzeichnis `config/` wechseln und Analyse laufenlassen mit:
  `sframe_main Cycle1_config.xml`
- Ausgabe betrachten:
  `root -l Cycle1.mc12_147771.e1434_s1499_s1504_r3658_r3549_p1328.root`
- Was enthält die Ausgabe?

Blick in den Code: `Cycle1_config.xml`

Grundlagen von XML (*Extensible Markup Language*)

Blick in den Code: `Cycle1_config.xml`

Grundlagen von XML (*Extensible Markup Language*)

- Gleiches Prinzip wie HTML (*Hypertext Markup Language*)
- Öffnende Tags `<bla>` und schließende Tags `</bla>` umfassen Blöcke von Text
  - Zusammenfassen falls kein enthaltener Text: `<bla />`
  - beliebiges Verschachteln möglich
- Attributzuweisungen: `<bla Attribut="Wert">`
- Kommentare: `<!-- blabla -->`

Blick in den Code: `Cycle1_config.xml`

- `<JobConfiguration JobName="AnalysisCycleDGJob" OutputLevel="INFO">`:
  gibt dem Job einen Namen, definiert welche Nachrichten ausgegeben werden (Ausgabelevel)
- `<In FileName="..." />`: Definition der Eingabedatei
- `<Item Name="..." Value="..." />`: Nutzerkonfiguration
  - Beispiel: `<Item Name="METName" Value="MET_Egamma10NoTau" />`
    → im C++-Code wird entsprechender Variable der Wert "MET_Egamma10NoTau" zugewiesen

Grundlagen von XML (*Extensible Markup Language*)

- Gleiches Prinzip wie HTML (*Hypertext Markup Language*)
- Öffnende Tags `<bla>` und schließende Tags `</bla>` umfassen Blöcke von Text
  - Zusammenfassen falls kein enthaltener Text: `<bla />`
  - beliebiges Verschachteln möglich
- Attributzuweisungen: `<bla Attribut="Wert">`
- Kommentare: `<!-- blabla -->`

Blick in den Code: `Cycle1.cxx`

- In `BeginInputFile()`:
  Objekte (Elektronen, Myonen, Jets, ...) mit entsprechenden Branches im D3PD verbinden:
  ```
  m_elec.SetPrefix("el_");
  m_elec.ReadFrom( inputTree );
  ```
  Danach stehen Variablen*) als Objekteigenschaften zur Verfügung.
- In `BeginInputData()`:
  Histogramme werden erstellt:
  ```
  h_n_muon = Book( SH1F( "h_n_muon",
      "Number of muons;n", 10, -.5, 9.5), "event");
  ```
- In `ExecuteEvent()`:
  Histogramme werden gefüllt:
  ```
  h_n_muon->Fill(m_muon.n());
  ```
- z. B. `m_muon[i].xxx()`:
  erlaubt Zugriff auf Eigenschaft “xxx” von Myon Nr. i

SFRAME: Eine Analyse mit Physikobjekten – Aufgaben

Aufgabe 0: Ausgabelevel

Ändern Sie die Ausgabelevel auf DEBUG.
(Weitere Möglichkeiten: VERBOSE, INFO, WARNING, ERROR, FATAL, ALWAYS)

Aufgabe 1: Mehr Statistik

Im in der Analyse verwendeten Datenordner liegen noch weitere Dateien. Fügen Sie einige oder alle von diesen der Konfiguration hinzu und lassen Sie den Code über den erweiterten Datensatz laufen.

Aufgabe 2: Berechnen von m_{ll}

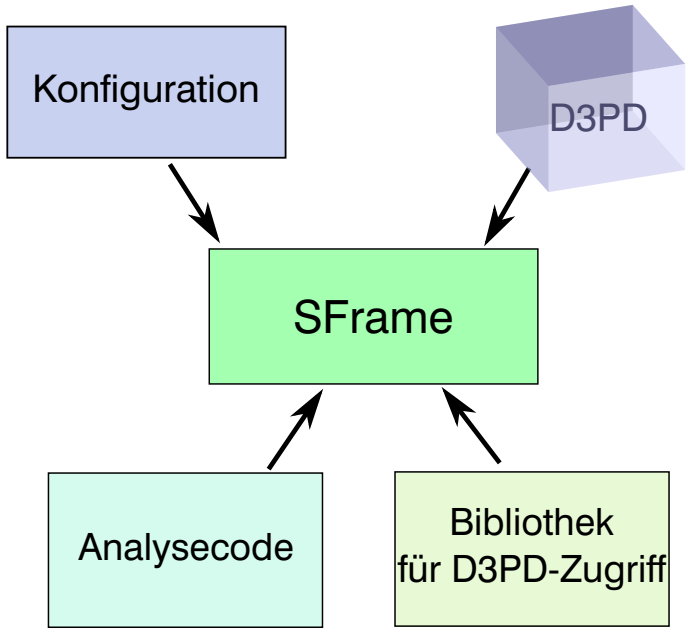
Verwenden Sie Ereignisse mit (mindestens) 2 Myonen, um die invariante Masse von Myonpaaren zu berechnen und in ein Histogramm einzutragen.

Aufgabe 3: XML-Konfiguration

Erweitern Sie die Berechnung von m_{ll}, so dass nur Myonen oberhalb einer gegebenen p_T-Schranke verwendet werden. Gestalten Sie den Code dabei so, dass diese Schranke über die XML-Konfigurationsdatei vorgegeben werden kann.

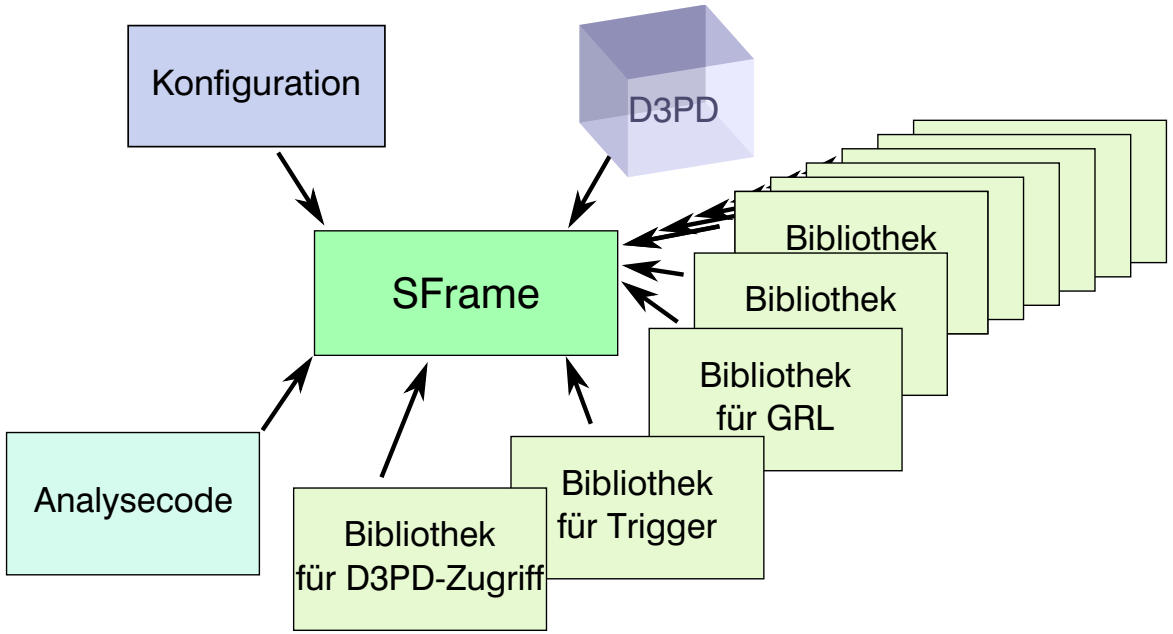
SUSY Tools

- Stellt eine Reihe von wichtigen Funktionen und Definitionen bereit, die in der ATLAS-Supersymmetrie-Gruppe Standard sind
- Auch die Objektdefinitionen, die für die Objektselektion verwendet werden
- Bindet dafür weitere Softwarepakete ein
  $\rightarrow$ diese meist für alle Physik-Analysen relevant (nicht nur SUSY)
- Paketmanager (für Kompilieren etc.) ist RootCore
- Wollen im folgenden einige Pakete davon verwenden



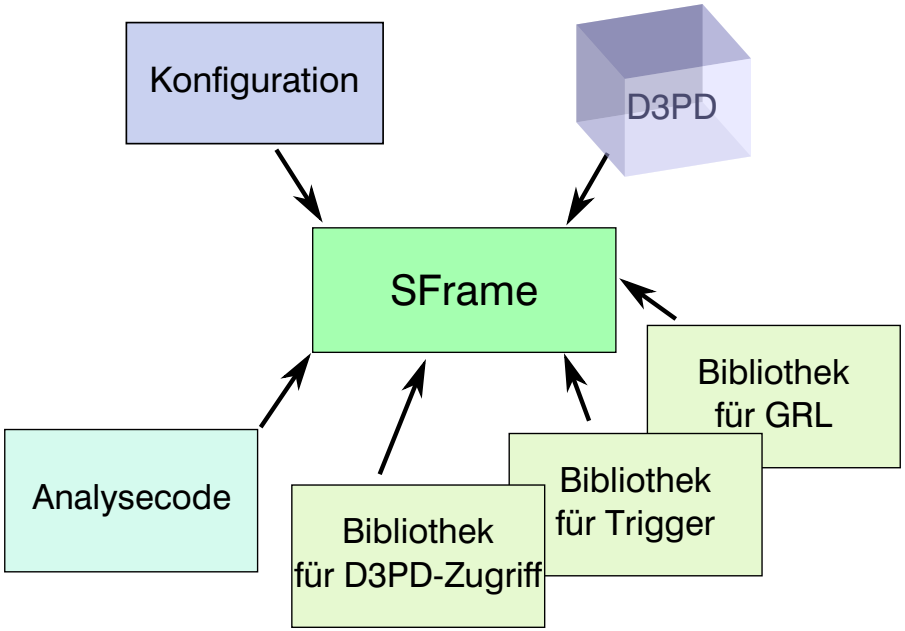
SUSY Tools

- Stellt eine Reihe von wichtigen Funktionen und Definitionen bereit, die in der ATLAS-Supersymmetrie-Gruppe Standard sind
- Auch die Objektdefinitionen, die für die Objektselektion verwendet werden
- Bindet dafür weitere Softwarepakete ein
  $\rightarrow$ diese meist für alle Physik-Analysen relevant (nicht nur SUSY)
- Paketmanager (für Kompilieren etc.) ist RootCore
- Wollen im folgenden einige Pakete davon verwenden



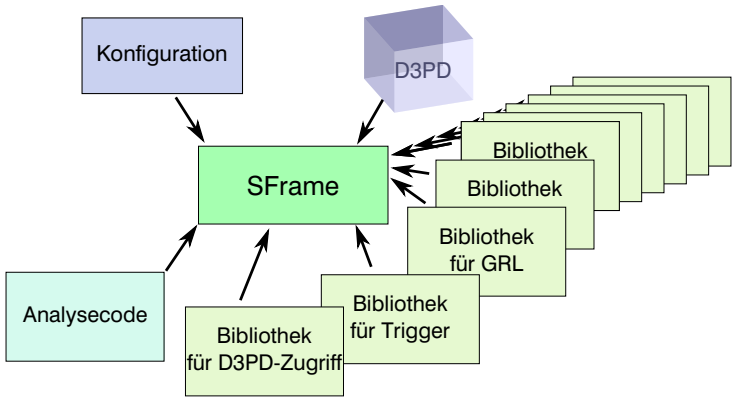
SUSY Tools

- Stellt eine Reihe von wichtigen Funktionen und Definitionen bereit, die in der ATLAS-Supersymmetrie-Gruppe Standard sind
- Auch die Objektdefinitionen, die für die Objektselektion verwendet werden
- Bindet dafür weitere Softwarepakete ein
  $\rightarrow$ diese meist für alle Physik-Analysen relevant (nicht nur SUSY)
- Paketmanager (für Kompilieren etc.) ist RootCore
- Wollen im folgenden einige Pakete davon verwenden



Installation

- Neue “saubere” Konsole öffnen
- Archiv mit Code in ein lokales Arbeitsverzeichnis kopieren und entpacken:
  - `/project/etp/Bachelor15/SFrame_RootCore.tgz`
  - (Kopieren: `cp [von] [nach]`)
  - (Entpacken: `tar zxf [Datei]`)
- Root Version `5.34.19` laden: `module load root/5.34.19`

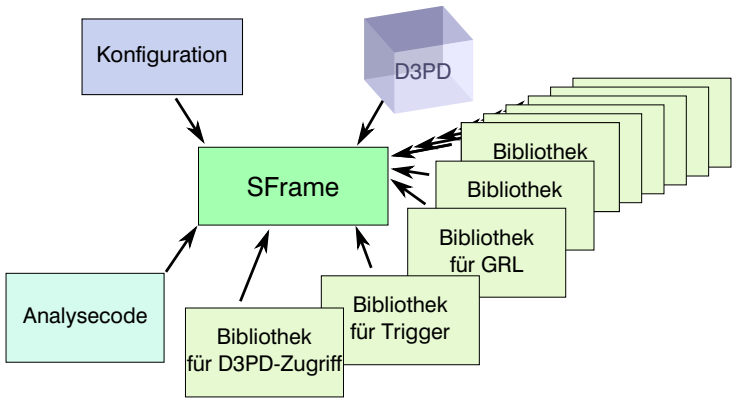


Installation

- In Unterverzeichnis `SFrame_RootCore/External/RootCore/` wechseln und die Pakete kompilieren:

```
./configure
source scripts/setup.sh
cd ..
$ROOTCOREDIR/scripts/find_packages.sh
$ROOTCOREDIR/scripts/compile.sh
```

  - (→ passt die Pfade von RootCore Ihrem Ausführungsverzeichnis an, Sie sehen das in `scripts/setup.sh`)
  - (das Kompilieren wird einige Minuten dauern – schauen Sie sich solange den Code an oder verwenden Sie die vorkompilierte Version in `/project/etp/Bachelor15/SFrame_RootCore_External_compiled.tgz`)



Setup

- Neue “saubere” Konsole öffnen (Zwischenschritt ohne sf.sh Skript war noetig da erst RootCore kompilliert werden musste und die Setupskripte fuer RootCore erstellt werden mussten mit Hilfe von `./configure`)
- In Verzeichnis `SFrame_RootCore/` wechseln und Setup starten:
  `source sf.sh`
- In Unterverzeichnis `Analyse1/` wechseln und die Analyse kompilieren:
  `make`
- In Unterverzeichnis `config/` wechseln und Analyse laufenlassen mit:
  `sframe_main Cycle1_config.xml`
- Die wesentliche Änderung ist die Verwendung der Triggerinformationen.
- Wo findet sie im Code statt? Welcher Trigger wird verwendet?

SFRAME: Kleine Projekte

Projekt 0: Fingerübungen

Fügen Sie ein “cutflow”-Histogramm zur Analyse hinzu, in dem die Anzahl verbleibender Ereignisse nach jedem Schritt eingetragen wird. Am besten, Sie verwenden alphanumerische Labels. Einfache Schnitte können auf dem Trigger basieren, den Primärvertizes, . . .

Projekt 1: Objektselektion

Schreiben Sie Funktionen, die Objekte anhand von Qualitätskriterien selektieren und den *overlap removal* durchführen. Für die Qualitätskriterien gibt es vordefinierte Funktionen in den SUSY-Tools (`Fill...()`). Sie können z. B. Vektoren verwenden, in denen die Indizes der selektierten Objekte gesammelt werden.

Projekt 2: Trigger-Effizienzen

Die Effizienz eines gegebenen Triggers ist definiert als die Anzahl der Ereignisse, in denen dieser feuert, dividiert durch die Gesamtzahl Ereignisse. Berechnen Sie die Trigger-Effizienz eines Myon-Triggers als Funktion des Offline-p_T der Myonen. Was stellen Sie fest? Welche Form der Effizienzkurve erwarten Sie, wenn die Auflösung des gemessenen p_T im Trigger gaußförmig um den offline gemessenen Impuls verteilt ist? Können Sie auch Effizienzen für andere Trigger bestimmen? ($\not{E}_T$-Trigger bieten sich an und sind in gewisser Weise noch einfacher zu studieren als Myon-Trigger. Warum?)

Projekt 3: *Truth*-Informationen

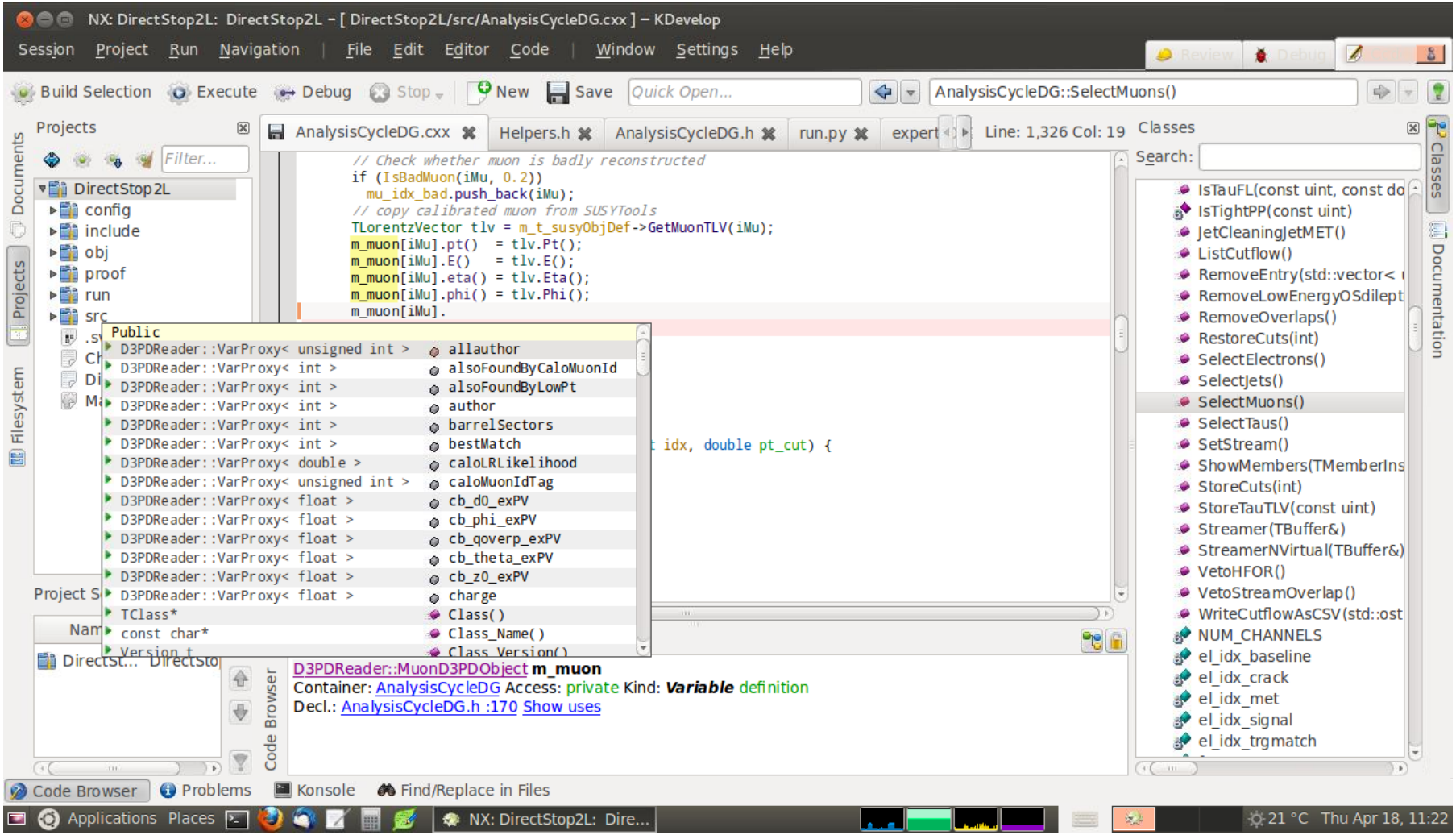
Die Monte-Carlo-Samples enthalten nicht nur die rekonstruierten Objekte, sondern auch eine Aufzeichnung des tatsächlichen (simulierten) Ereignisses. Versuchen Sie, die rekonstruierten Myonen mit den simulierten in Einklang zu bringen und bestimmen Sie die Auflösung des rekonstruierten p_T in Relation zum wahren p_T. (Hinweis: Verwenden Sie das `D3PDReader::TruthParticleD3PDObject`-Objekt, um auf diese Informationen zuzugreifen.)

Projekt 4: Arbeiten mit "echten Daten"

Bisher haben wir mit simulierten Daten gearbeitet. In `data12_8TeV.00206573.physics_Muons.merge.NTUP_SUSYSKIM.r4065_p1278_p1328_p1329_tid01172369_00` finden Sie einige Dateien aus einem "echten" Datensatz aus dem `Muons`-Datenstrom. Wenden Sie die Berechnung der invarianten Masse hierauf an und vergleichen Sie mit den simulierten Daten. Beachten Sie, dass Sie hierfür eine GRL und eine Triggerselektion (z. B. basierend auf EF_mu24i_tight) verwenden müssen. Was passiert, wenn Sie keinen (Myon-)Trigger verwenden?

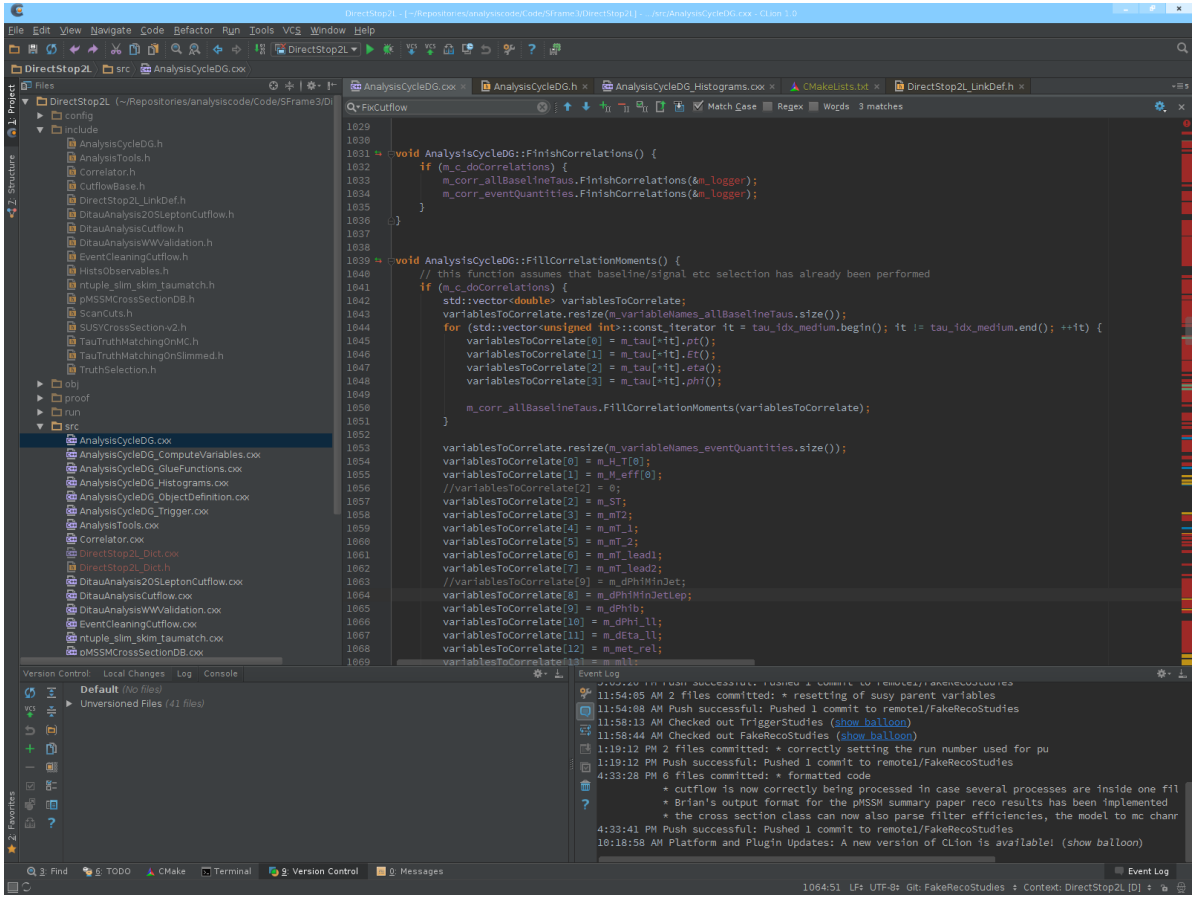
Backup

Werkzeuge für Entwickler: KDevelop



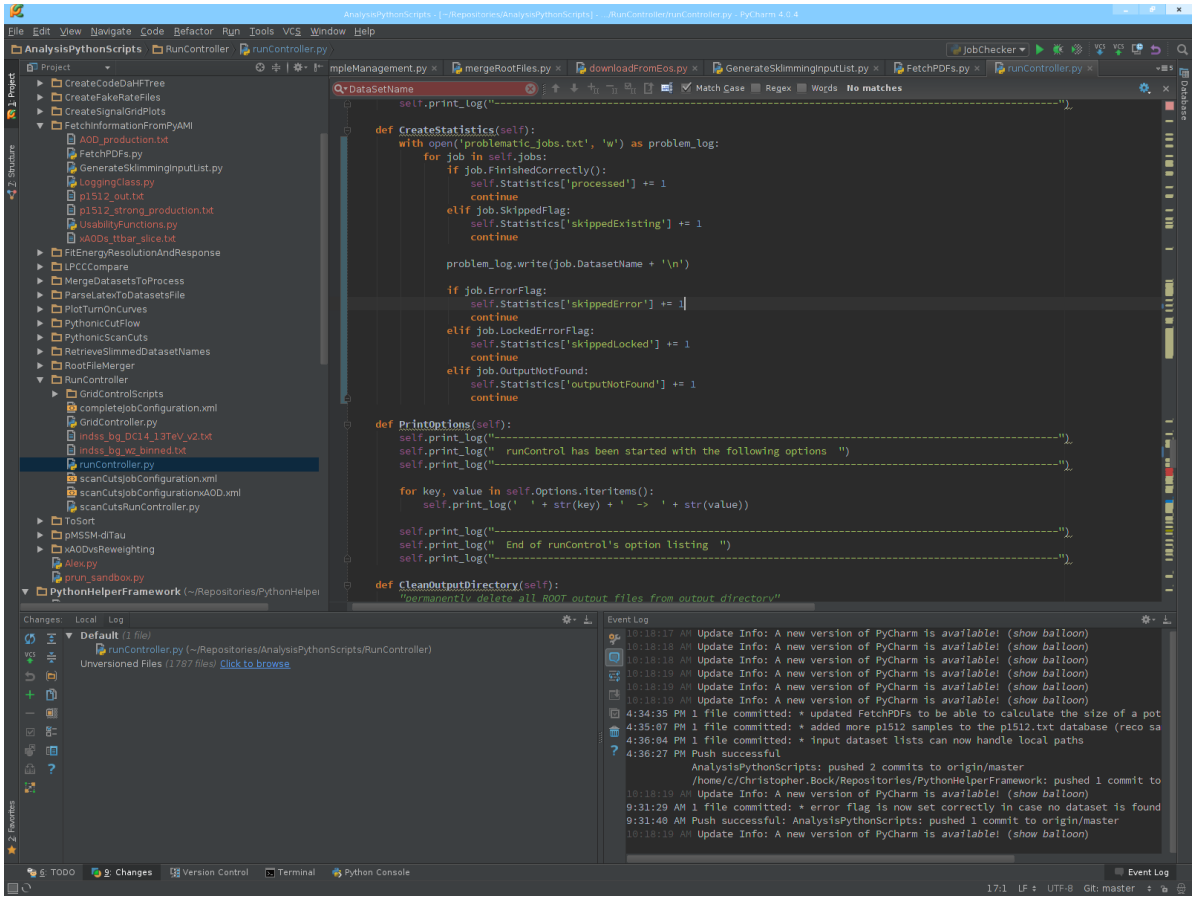
- Vorschläge zur Codevervollständigung (insbesondere in Verbindung mit D3PDReader-Klassen hilfreich!)
- Syntaxhervorhebung, Liste der Klassen, wo wird was deklariert / verwendet etc.

Werkzeuge für Entwickler: CLion



- Vorschläge zur Codevervollständigung (insbesondere in Verbindung mit D3PDReader-Klassen hilfreich!)
- Syntaxhervorhebung, Liste der Klassen, wo wird was deklariert / verwendet etc.
- Funktionen zur Refaktorierung von Code
- Integrierte Versionskontrolle

Werkzeuge für Entwickler: PyCharm



- Vorschläge zur Codevervollständigung in **Python**
- Syntaxhervorhebung, Liste der Klassen, wo wird was deklariert / verwendet etc.
- Funktionen zur Refaktorierung von Code
- Integrierte Versionskontrolle

PROOF

Was ist PROOF?

- PROOF = Parallel ROOT Facility
- Erweiterung von ROOT zum parallelen Rechnen auf mehreren Prozessoren oder Computerverbünden
- Durch die Unabhängigkeit von aufgezeichneten Detektorkollisionen (Ereignissen) $\Rightarrow$ einfache Parallelisierung der Analyse möglich
- “PROOF-lite” erlaubt Ausprobieren auf einem einzigen Rechner
- Oft ist “Flaschenhals” nicht Rechenleistung (CPU), sondern Einlesen der Daten $\rightarrow$ Verteiltes Rechnen auf mehreren Knoten / Rechnern $\Rightarrow$ höhere verfügbare Bandbreite
- Analysecode: über “PAR”-Pakete auf die entfernten Rechner verteilt
- On-demand Verbünde können mithilfe von “PoD” (PROOF-on-demand) recht einfach auf den Rechnern unseres Institus erstellt werden, siehe ETP Wiki: Dynamic Proof Cluster
- Weitere Informationen: Homepage von PROOF

SFrame und PROOF (optional)

- Ändern Sie in der XML-Konfiguration den Modus (`RunMode`) von `"LOCAL"` auf `"PROOF"`
- PROOF benötigt die Eingabedateien als registrierten PROOF-Datensatz
  - Registrieren Sie einen PROOF-Datensatz, der die bisher verwendeten Eingabedateien enthält (Hinweis: folgen Sie den Anweisungen auf http://root.cern.ch/drupal/content/working-data-sets#register)
  - Überprüfen Sie, ob die Registrierung erfolgreich war (Hinweis: verwenden Sie `TProof::ShowDataSets()`)
  - Tragen Sie diesen in der XML-Konfiguration anstelle der Dateien ein (Hinweis: verwenden Sie das Tag `<DataSet Name="..." Lumi="1.0" />`)
- Starten Sie SFRAME wie gewohnt
  - (Hinweis: zum Ausprobieren empfiehlt sich `SFrame_work1.tgz` (ohne ROOTCORE...))
  - (Hinweis: andernfalls: `RootCore.par` → erzeugen mit `make_par.sh` in `scripts/`-Ordner von ROOTCORE, verlinken in `SFrame/lib/`, Pakete in XML-Konfiguration einbinden)
- Was ist anders? Was ist gleich? Wann lohnt sich die Verwendung von PROOF?

Cheat Sheets

Cheat Sheet

Linux: Important Places in the File System

- `/home/your.name/`: your home directory, hint: can be abbreviated by `~`
- `/project/etp/Bachelor15/`: directory with these slides and additional material
- `.`: refers to current directory
- `..`: refers to one directory up
- `/tmp/`: temporary directory, available on all machines. Here you can play around and try out stuff; Files will get deleted after some time (usually $\lesssim 14$ days).

Linux: File Operations

- `pwd`: print current working directory
- `ls -l`: list contents of current directory (`-l` enables long format)
- `mkdir name`: creates a subdirectory named `name` in the current directory
- `cp from to`: copies file named `from` to a file named `to` (in current directory)
- `cp -r dirfrom/ dirto/`: copies directory `dirfrom/`, and all files (and subdirectories recursively) contained in that, into directory `dirto/`. Note: If `dirto/` existed before, the copy of `dirform/` is created as a subdirectory of `dirto/`.
- `rm name`: deletes file `name`.
  Note: there is no “undelete” on Linux.
- `rmdir name/`: deletes directory `name`, note: directory must be empty

ROOT: Starting ROOT

- `root -l`: opens ROOT
- `root -l name.root`: opens ROOT and makes file `name.root` available

ROOT: Within ROOT

- `new TBrowser();`: opens TBROWSER.
  You’d find the file `name.root` in the tree on the left near the top under “ROOT files”.
  - double-click on file to open it
  - double-click on tree name to expand and collapse list of tree’s contents
  - double-click on branches or histograms to draw them
- Drawing and editing histograms:
  - Zoom-in: click and hold on one of the histogram axes to select range you want to be shown
  - Unzoom: right-click on one of the histogram axes and select `UnZoom` from the pop-up menu
  - Logarithmic scale: right-click close to the border of the canvas (= histogram drawing area) and select `SetLogx` from the pop-up menu
  - Grid: right-click close to the border of the canvas (= histogram drawing area) and select `SetGridx` from the pop-up menu
  - Colored 2d plots: enter `COLZ` in the field labelled “Draw Option:” (before double-clicking to draw the histogram)
  - Overlaying histograms: enter `SAME` in the field labelled “Draw Option:” (before double-clicking to draw the second, third, . . . histogram)

Linux: More Useful Commands

- `ssh -Y gar-ws-etpNN`: log-in to workstation in our cluster.
  Replace `NN` with the number of the node, and be polite when using other people's computers.
  `-Y` enables graphic user interface (e. g. for the use of TBROWSER).
- `exit`: close connection to workstation / exit terminal.
- `grep text filename(s)`: find text in files
- `less name`: list contents of file `name`.
  Use arrows to go up and down, press `q` to exit.
- `tar zxf file.tgz`: unpacks the TAR archive `file.tgz` in the current directory.
  Note: Be careful, existing files with the same names as files contained in the TAR archive will be overwritten without notice. Might want to do this is an empty directory.
- `[TAB]`: the `[TAB]` key auto-completes the current command based on available commands and files in the current directory or path you have started to type in. If nothing happens, try hitting `[TAB]` again (twice) to list suggestions (e. g. if there's more than one possible completion).
  Very useful e. g. to not have to type in long filenames!

Problembehebung

- *Loading of xml document "Cycle1_config.xml" failed*
  - Existiert die Konfigurationsdatei? Vielleicht vergessen, ins `config/`-Verzeichnis zu wechseln?
- *failed to load external entity "JobConfig.dtd"* und jede Menge “validity error”:
  - Die Struktur der Konfigurationsdatei ist in der Datei JobConfig.dtd festgelegt – diese muss im Ausführungsverzeichnis von SFRAME vorhanden sein.
- *sframe_main: Befehl nicht gefunden.*
  - `source sf.sh` gemacht?
- *parser error : Opening and ending tag mismatch*
  - In der Konfigurationsdatei passen öffnende (“<...>”) und schließende (“</...>”) Tags nicht zusammen.