

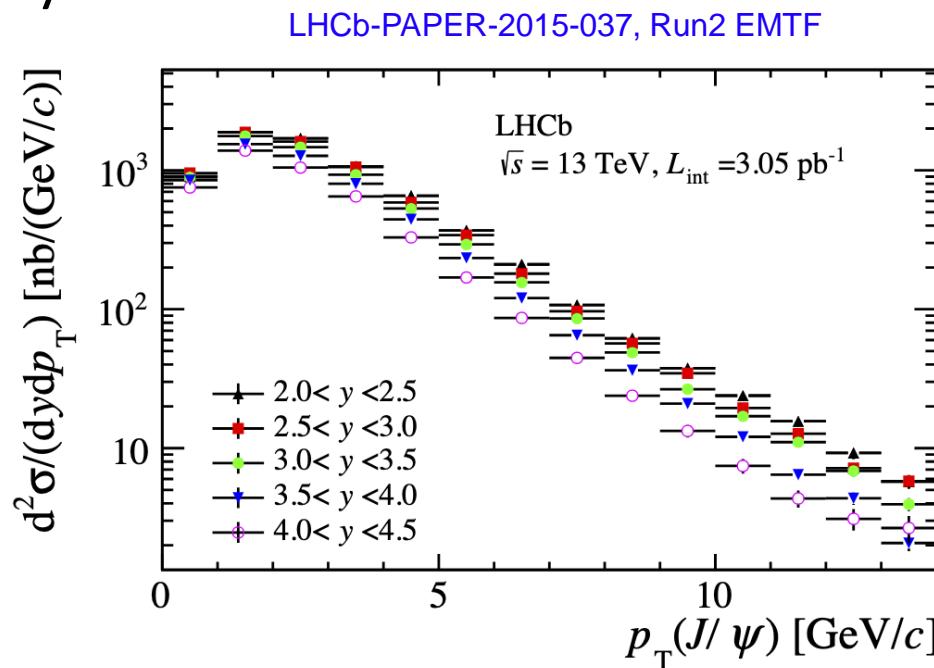
J/ψ cross-section measurement using Early Run3 data

The J/ψ Xsec task force team

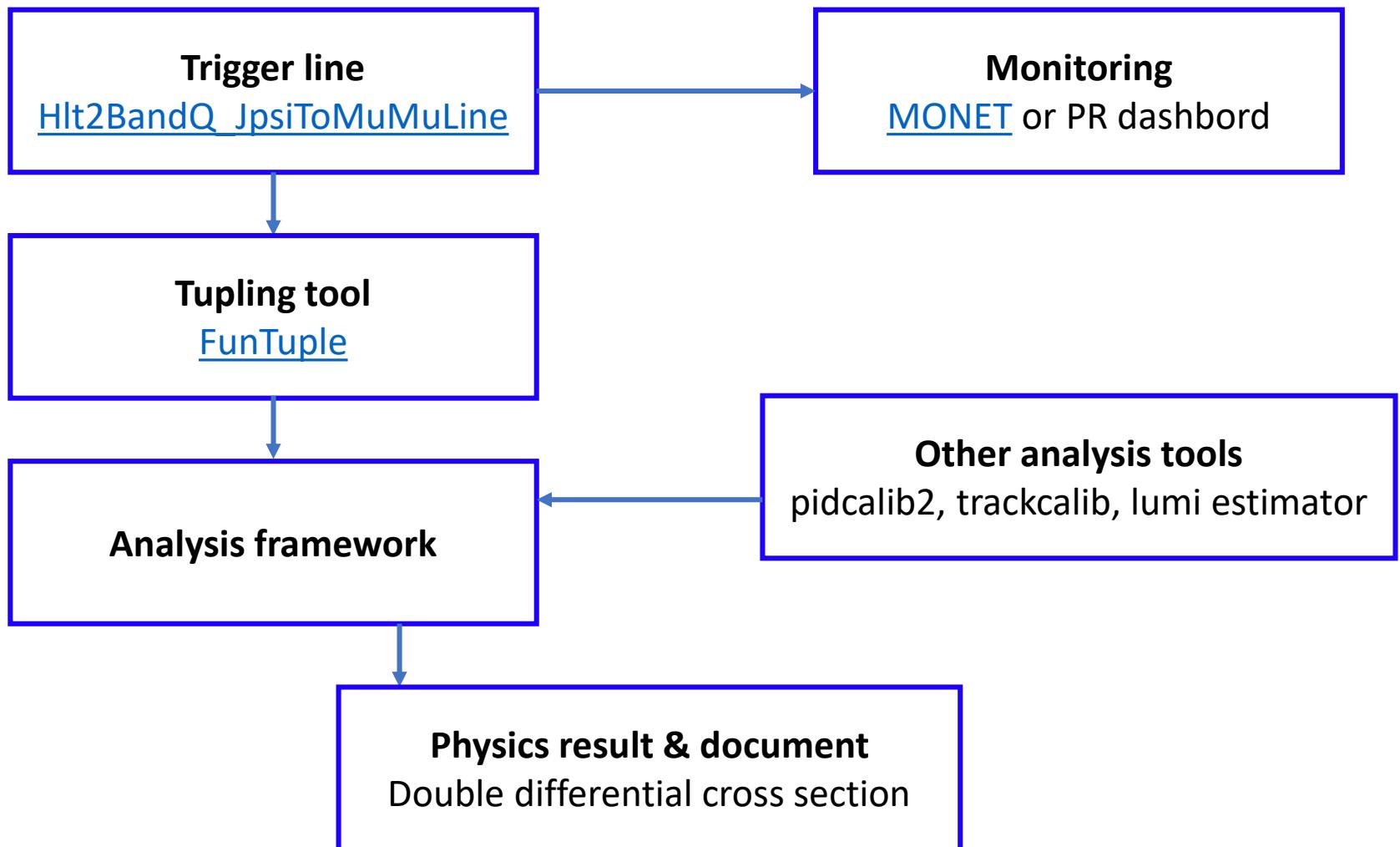
2022/03/14

Motivation

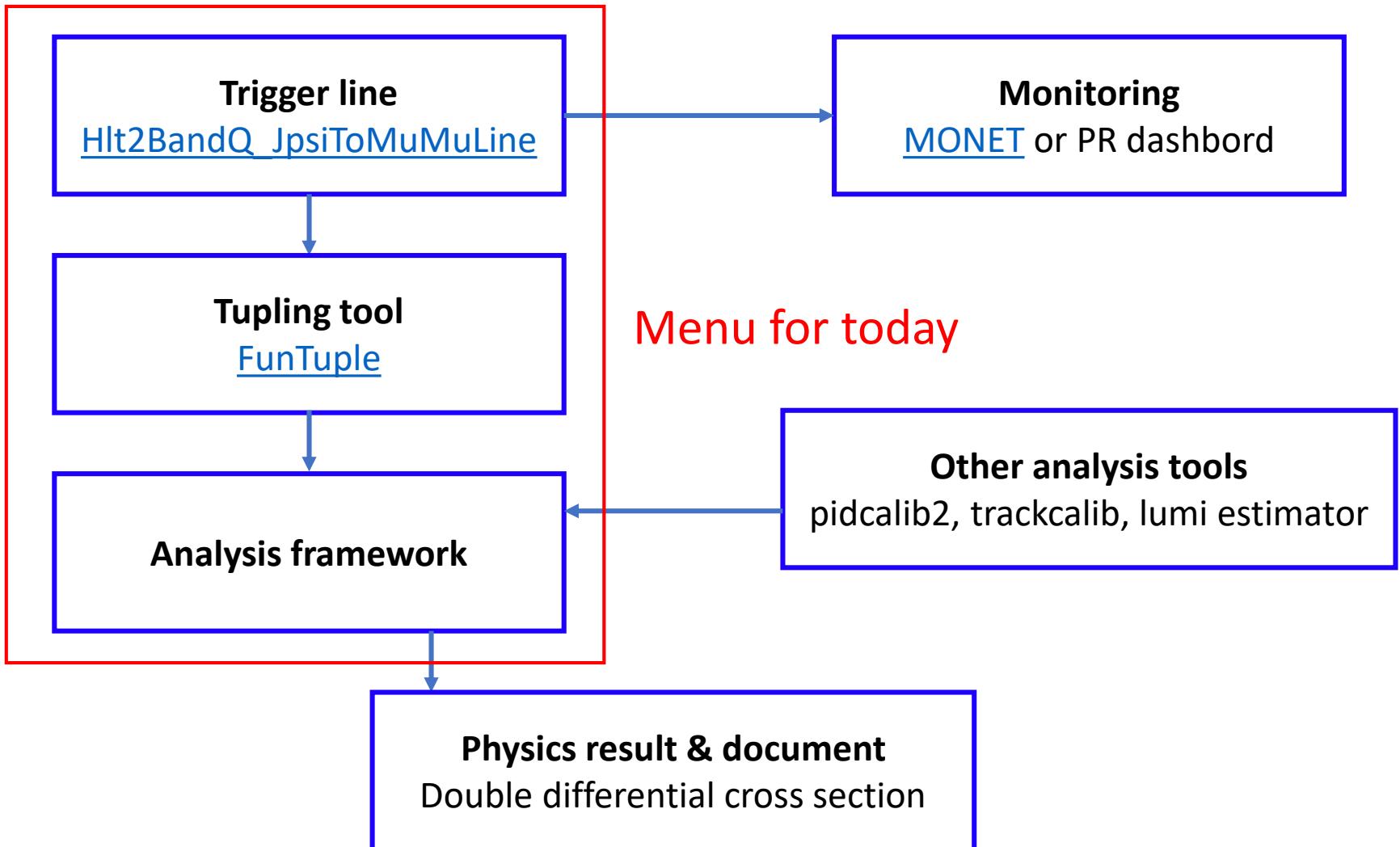
- Measure **differential & integrated J/ψ cross section** with new LHCb detector @ 13.8 TeV
 - Golden channel $J/\psi \rightarrow \mu^+ \mu^-$
 - Test Run3 performance of **trigger/tracking/PID/lumi/Sim**
 - Directly test **dimuon reconstruction**, essential for many B&Q analyses



Framework of the task force



Framework of the task force



Trigger line

- Similar as Run2 prompt $J/\psi \rightarrow \mu^+ \mu^-$ line
 - Cut on PID and p_T of muon
 - Only persist J/ψ candidate to save bandwidth

Variable	Requirement
$p_T(\mu)$	> 300 MeV
PID of μ	isMuon, PIDmu > -5
χ^2_{vtx}	< 25
Mass window of J/ψ	PDG mass ± 120 MeV

Will change to ± 150 MeV [MR]

- Efficiency (no hlt1 filtering):
 - ~30% for prompt J/ψ sample
 - ~50% for J/ψ from b decays ($B^+ \rightarrow J/\psi K^+$)

Tupling tool

- FunTuple: a new tupling tool for Run3 analysis
- New features (from user's point of view):
 - Branches to save are customized and directly visible

Run2 Tuple tool

```
tupletools = ["TupleToolGeometry",
              "TupleToolKinematic",
              "TupleToolEventInfo",
              "TupleToolPid",
              "TupleToolTrackInfo",
              "TupleToolRecoStats"]
```

Convenient when coding DaVinci
90% branches are not used

- Support ThOr functors

FunTuple

```
jpsi_variables = FunctorCollection({
    "ID": F.PARTICLE_ID,
    "KEY": F.OBJECT_KEY,
    "PT": F.PT,
    "PX": F.PX,
    "PY": F.PY,
    "PZ": F.PZ,
    "M": F.MASS,
    "ENERGY": F.ENERGY,
    "P": F.P,
    ....
```

Reduce tuple size, easy to handle
Extra effort to check if the list is complete
for an analysis

Tupling tool

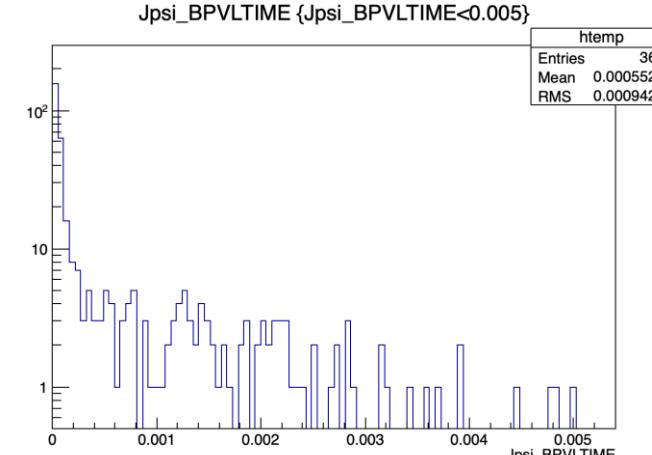
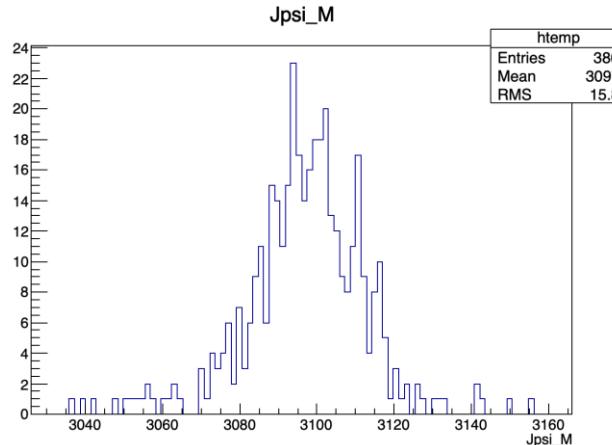
- A “prototype” FunTuple [script](#) available
 - Directly read Hlt2 output

```
jpsi2mumu_data = make_data_with_FetchDataFromFile(  
    "/Event/HLT2/Hlt2BAndQ_JpsiToMuMuLine/Particles")
```

- Apply loose preselections

```
code = require_all( "M > 3036.", "M < 3156." )  
sel_jpsi2mumu_data = ParticleFilter(particles=jpsi2mumu_data, name = "sel_jpsi", Code=code)
```

- Dump basic information for a Xsec analysis



Some event-level
information not
available now,
covered by [[this issue](#)]

MC sample preparation

- **3 Million** inclusive $J/\psi \rightarrow \mu^+ \mu^-$ events requested
 - Under production, ~10% available
 - **XDIGI output.** Central Reco not available for now
- Run Moore: XDIGI → DST
 - [[Option file](#)] available for running Hlt2 Reco without UT
 - Use ganga to speed up the production [[tutorial](#), [example](#)]
 - **Time consumption**
 - ~1000 events/hour/subjob
 - **3Million events: 15 hours if split into 200 subjobs.** Acceptable
 - **Size of output files**
 - ~1.5GB/1000 events. **4.5TB/3Million events.** Need a few rounds
- Run DaVinci: DST → ROOT
 - Much faster than reconstruction. Plan to do it locally

Analysis framework

- Under construction in parallel with tuple preparation
- Tools for a more flexible framework:
 - Snakemake: defines the input & output for each step, visualize the analysis logic, easy to rerun the analysis chain

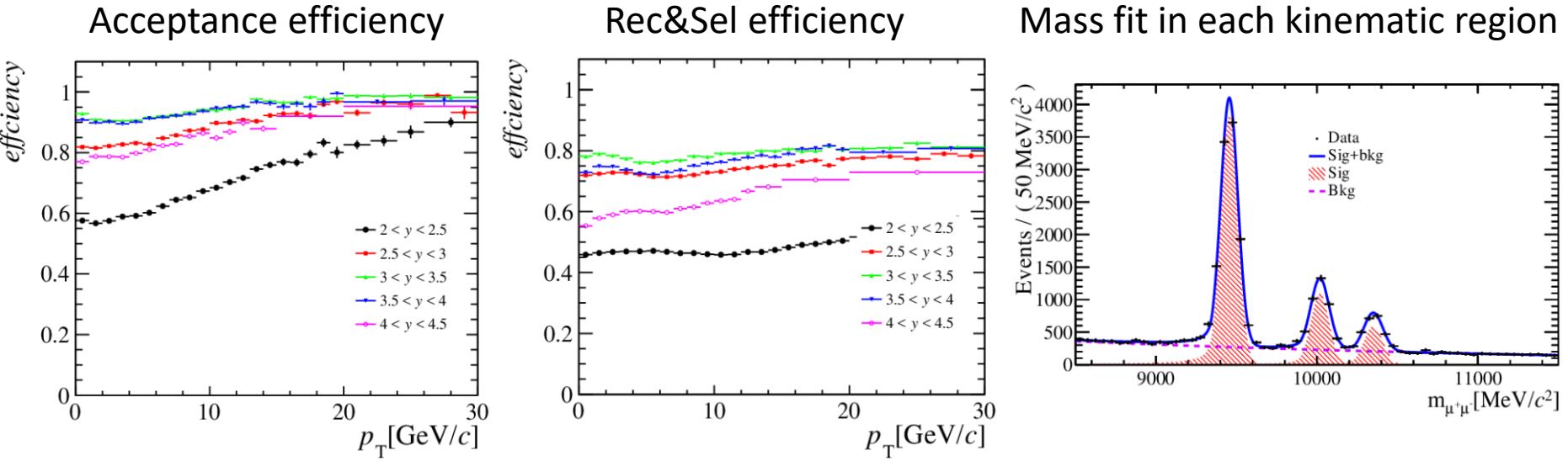
```
rule eff_acc:  
    input:  
        inputsrc = "scripts/eff_acc/GetEff.py" ,  
        inputfile1 = dir_prefix + "/gen_mc/{state}u.root",  
        inputfile2 = dir_prefix + "/gen_mc/{state}d.root"  
    output:  
        outputfile = dir_prefix + "/result/eff_hist_root/eff_acc{state}.root",  
        outputfig = dir_prefix + "/figure/eff_acc/eff_acc{state}.pdf"  
    shell:  
        "python3.6 {input.inputs} -i {input.inputfile1} -i {input.inputfile2} -o {output.outputfile} -of {output.outputfig} "
```

- Yaml: well separate the configurations from algorithms

```
Ups_cut:  
    mass_cut: "abs(ups_MM-10000.)<1500."  
    vertex_prob: "ups_ENDVERTEX_CHI2<TMath::ChisquareQuantile(0.995,1)"  
    pv_constraint: "ups_PVDTF_CHI2NDOF<5"  
Global_cut:  
    onepv: "nPV==1"
```

Analysis framework

- External inputs (temporary)
 - 2018 $\gamma \rightarrow \mu^+ \mu^-$ data & MC
 - As a in-parallel effort for the AP of Run2 γ production analysis
 - Run 2 PIDCalib & TrackCalib tables
- What we can get now by typing `snakemake`



Status of the analysis framework

- Status for each modules
 - **Selection**: preliminary version implemented
 - **Signal-yield extraction**: not ready
 - For now a Υ mass fit. To be replaced by J/ψ mass- t_Z 2D fit
 - **Efficiency estimation**: preliminary version implemented
 - **Systematic uncertainty**: not implemented yet
- Will give a more detailed report when updated using Run3 J/ψ MC samples

Analysis framework

- The final version should contain:

3 Candidate reconstruction and selection	3
3.1 Trigger and Turbo stream selection	3
3.2 Offline selection refinement	4
3.2.1 MuonID strategy	5
4 Cross-section determination	6
5 Signal extraction	7
5.1 Determination of the prompt and detached signal yields	9
6 Efficiency determination	12
6.1 Geometrical acceptance	13
6.2 Reconstruction-selection efficiency	13
6.3 Muon identification efficiency	14
6.4 Trigger efficiency	15
6.5 Total efficiency	15
7 Systematic uncertainties	15
7.1 Signal extraction	16
7.1.1 Signal mass shape	16
7.1.2 Fit to t_z	17
7.2 MuonID efficiency	17
7.3 Tracking efficiency	18
7.4 Offline selection efficiency	21
7.5 Trigger efficiency	22
7.6 Unknown spectrum	24
7.7 Polarization scenarios	25
7.8 Other systematic uncertainties	26

Offline Event selection

mass and lifetime fit, to extract
signal prompt/from-b J/ψ yields
in each kinematic region

Efficiency estimation in each
kinematic region

Systematic uncertainty studies

Conclusion

- Plan to measure differential & integrated J/ψ cross section with new LHCb detector @ 13.8 TeV
- Brief summary of the status:
 - Hlt2 line ready
 - Preliminary version of tupling tool ready
 - Started to develop analysis framework
 - A more careful tuning will be performed when Run3 J/ψ MC sample available

Thank you for your attention !
Any questions or comments ?