



# Implementing new analyses with ADL/CutLang

Sezen Sekmen, Gökhan Ünel, Changgi Huh

Analysis Description Language Tutorial & Hackathon  
21-22 Nov 2022, Kyungpook National University, Center for HEP



# Translating analyses to ADL

Primary goal: Everyone writes a new analyses with ADL syntax (can also work in groups 2 or 3)

- Let's focus on translating the analysis algorithm into ADL.
- Assess the feasibility and practicality of ADL.
- Test the **capacity of ADL syntax** in expressing different analyses. Spot **syntax limitations, issues**.
- Just write **what you can!**

Later stage: Running with CutLang:

- Once analysis ADL files are mature, we can **obtain relevant event files and run the analyses** using CutLang.
- We can even attempt **validating the analyses** using information provided by experiments.

This exercise is not limited to the timescale of the workshop!



# Notes for beginners

If you are new to the world of analysis:

- We can suggest a **simple analysis** for you from [our analysis list](#).
- For objects, use **simple 4-momenta and charge**.  
(objects are usually found in the “Event reconstruction” section of a paper, but sometimes definitions are seen in the “Event selection” sections.)
- Focus on **signal regions** first  
(usually found in the “Event selection” or “Analysis Strategy” sections.)
  - If it goes well, continue with **control and validation regions** (usually defined in the “Background estimation” section of a paper).



# Notes for experimentalists

If you are an experimentalist performing an analysis:

- Translate **your own analysis** into ADL. You know it best!
- For objects, use the **exact object attributes in your input data** (electron ID, b-tag discriminant name etc.) as you would in your analysis code.
- You could write **all selections, including signal, control and validation regions**.
- If the analysis has a **complex variable** or a **numerical variable** that needs an external function, write a **placeholder**.  
e.g. for “myfancyfunction” which takes in jets and MET, write  
`define myfancyvariable = fmyfancyfunction(jets, MET)`
- Let us know if you come across a **syntax limitation**, i.e. when ADL syntax is not sufficient to express your analysis. **What syntax would you use for such a case?**



# Notes for phenomenologists

If you are a phenomenologist or interested in reinterpretation:

- Translate **an analysis you may like to reinterpret**, or we can select from [our analysis list](#).
- For objects, use **4-momenta and charge**. If the publication states an object property (e.g. tightly identified electrons are selected), note this as a comment in the object block.
  - If the experiment has provided **efficiency maps**, please note it in the object block. Or maybe attempt making a **table to use with hit-and-miss method!** (as in [ex10\\_tableweight.adl](#))
- Focus on writing the **signal regions**.
  - If there are **cutflows** or **data and BG counts**, maybe attempt incorporating them into the ADL file as in [ex12\\_counts.adl](#) .
- If the analysis has a **complex variable** or a **numerical variable** that needs an external function, write a **placeholder**.  
e.g. for “myfancyfunction” which takes in jets and MET, write  
**define myfancyvariable = fmyfancyfunction(jets, MET)**
- Let us know if you come across a **syntax limitation**, i.e. when ADL syntax is not sufficient to express your analysis. **What syntax would you use for such a case?**



# Documenting and discussing

We can share our implementations in the [ADLAnalysisDrafts github repository](https://github.com/ADL4HEP/ADLAnalysisDrafts):  
<https://github.com/ADL4HEP/ADLAnalysisDrafts>

Please provide your [github username](#) to Sezen if you would like to join, or send your analysis ADL file directly on mattermost or via email.

Let's use mattermost to discuss questions and issues!