Merging idea

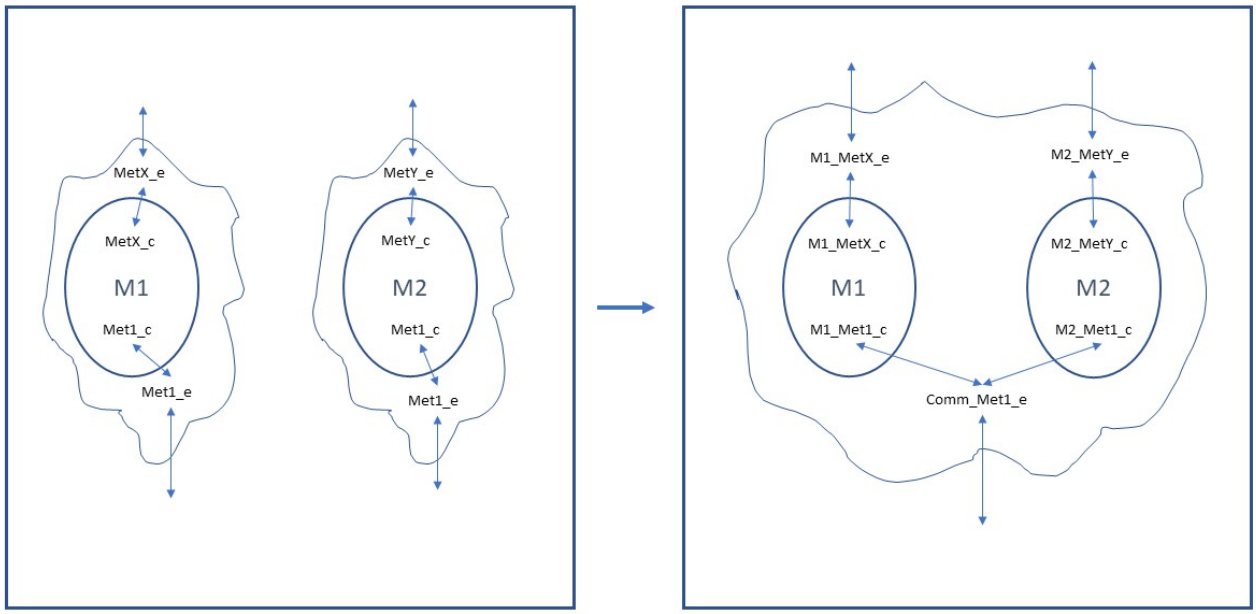
Draft of abstract:

Constraint-based reconstruction and analysis (COBRA) tools are of paramount importance for the study of metabolic interactions between organisms.
Several models and analytical tools that allow to simulate and study metabolic interactions between organisms of different species have been built, allowing an integrated analysis of multi-omics datasets of the different organisms involved in the interactions. The high diversity of organisms [give a list of modeled organisms] covered by these modeling techniques didn't come without limitations.
Various modelling platforms have been developed [citations], many of them being based on different databases, often translating to a limited ability of analyzing the interactions among them:

- This hampers the potential of the models (many more interactions could be studied if the models were compatible)
- Models can be readapted in a way that they can be "merged", but this is done manually and require extensive manual curation work
- A tool that automatizes these steps would facilitate the application of COBRA modelling to study many more multi-species metabolic interactions, allowing to fully employ the power of such analytical tools

Merging idea - Naming scheme

In order to merge 2 models *M1* and *M2*, we plan to first rename all the metabolites and reactions.
Let *identifier* be the ID of **any metabolite or reaction** from model 1, then its ID in the merged model will be *M1_identifier*.
We can distinguish 2 types of external metabolites, those who are common between the 2 models and those who are not. **Any external metabolite, both common and uncommon** will end up in a single common compartment of the merged model. The ID of common metabolites having *metID* as ID in the original model, will have an ID of the form *Comm_metID_e* in the merged model. The ID of the non-common external metabolites, being present for example only in *M1*, will be of the form *M1_metID_e* instead. This prefix and postfix convention allows to know if a metabolite is shared or not while keeping the postfix *"_e"*, which is used internally by cobrapy to detect external metabolites. In the same way, the exchange reactions for shared and not shared metabolites will be identified by *EX_Comm_metID_e* and *EX_M1_metID_e*


MetX_e
MetX_c
M1
Met1_c
Met1_e
MetY_e
MetY_c
M2
Met1_c
Met1_e
M1_MetX_e
M2_MetY_e
M1_MetX_c
M1
M1_Met1_c
M2_MetY_c
M2
M2_Met1_c
Comm_Met1_e