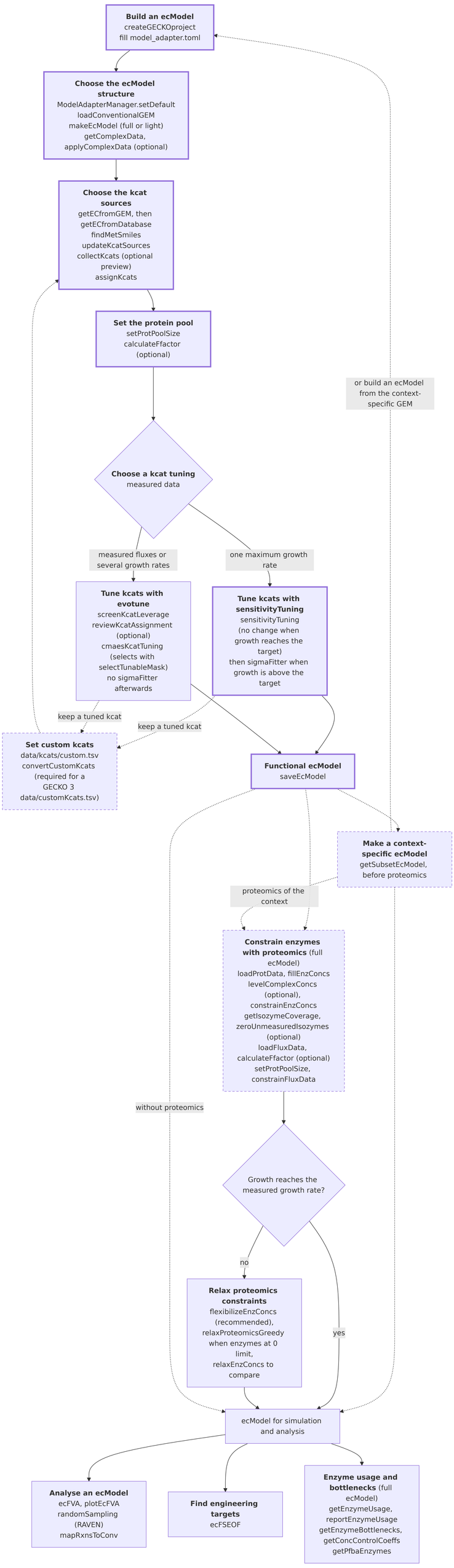

Build an ecModel
createGECKOproject
fill model_adapter.toml
Choose the ecModel structure
ModelAdapterManager.setDefault
loadConventionalGEM
makeEcModel (full or light)
getComplexData,
applyComplexData (optional)
Choose the kcat sources
getECfromGEM, then
getECfromDatabase
findMetSmiles
updateKcatSources
collectKcats (optional preview)
assignKcats
Set the protein pool
setProtPoolSize
calculateFfactor (optional)
or build an ecModel from the context-specific GEM
Choose a kcat tuning
measured data
measured fluxes or several growth rates
one maximum growth rate
Tune kcats with evotune
screenKcatLeverage
reviewKcatAssignment (optional)
cmaesKcatTuning
(selects with selectTunableMask)
no sigmaFitter afterwards
Tune kcats with sensitivityTuning
sensitivityTuning
(no change when growth reaches the target)
then sigmaFitter when growth is above the target
keep a tuned kcat
keep a tuned kcat
Set custom kcats
data/kcats/custom.tsv
convertCustomKcats
(required for a GECKO 3 data/customKcats.tsv)
Functional ecModel
saveEcModel
Make a context-specific ecModel
getSubsetEcModel,
before proteomics
proteomics of the context
Constrain enzymes with proteomics (full ecModel)
loadProtData, fillEnzConcs
levelComplexConcs (optional),
constrainEnzConcs
getIsozymeCoverage,
zeroUnmeasuredIsozymes (optional)
loadFluxData,
calculateFfactor (optional)
setProtPoolSize,
constrainFluxData
without proteomics
Growth reaches the measured growth rate?
no
yes
Relax proteomics constraints
flexibilizeEnzConcs (recommended),
relaxProteomicsGreedy
when enzymes at 0 limit,
relaxEnzConcs to compare
ecModel for simulation and analysis
Analyse an ecModel
ecFVA, plotEcFVA
randomSampling (RAVEN)
mapRxnsToConv
Find engineering targets
ecFSEOF
Enzyme usage and bottlenecks (full ecModel)
getEnzymeUsage,
reportEnzymeUsage
getEnzymeBottlenecks,
getConcControlCoeffs
getPfbaEnzymes