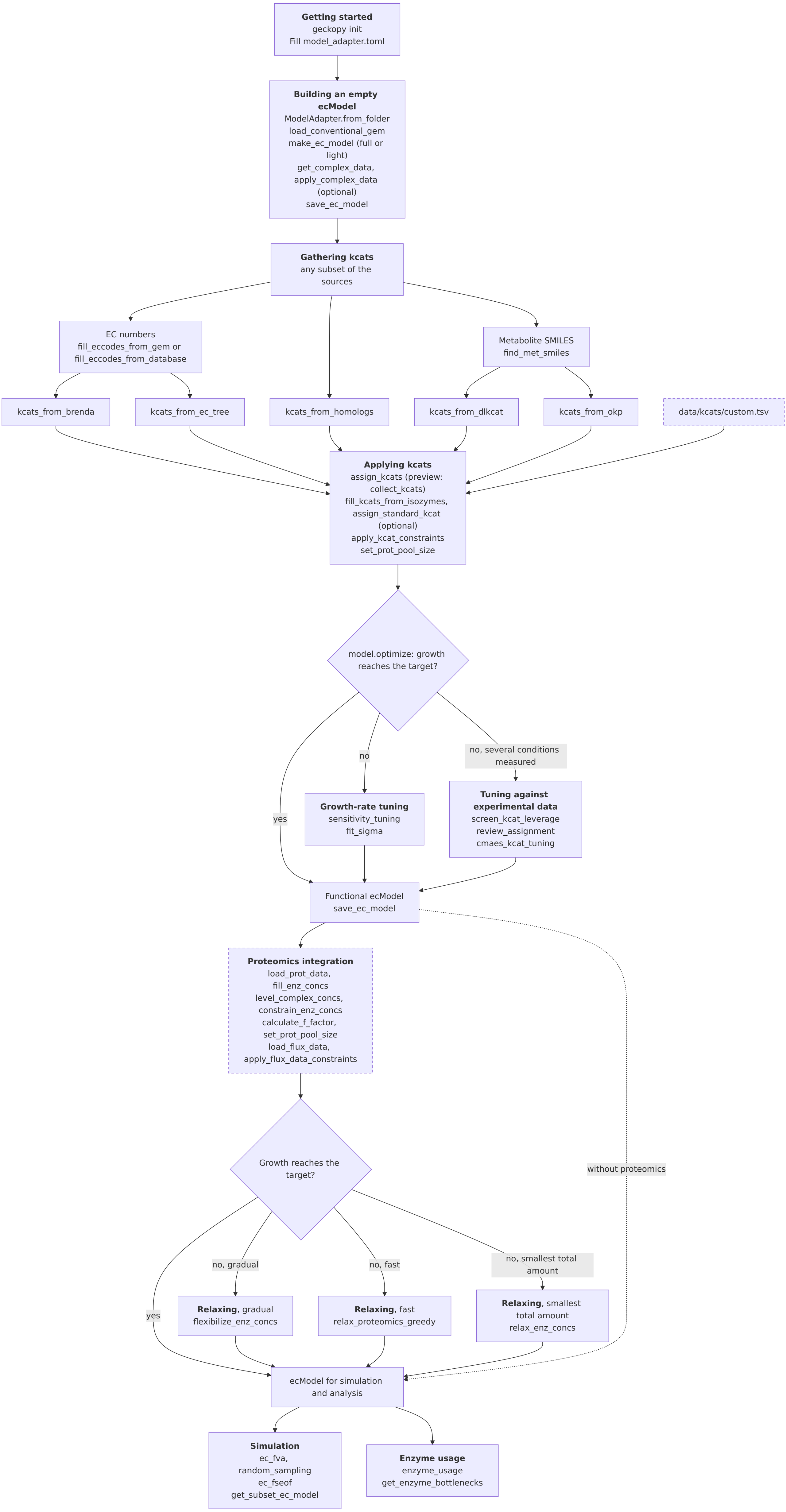

**Getting started**
geckopy init
Fill model_adapter.toml
**Building an empty ecModel**
ModelAdapter.from_folder
load_conventional_gem
make_ec_model (full or light)
get_complex_data,
apply_complex_data (optional)
save_ec_model
**Gathering kcats**
any subset of the sources
EC numbers
fill_eccodes_from_gem or
fill_eccodes_from_database
Metabolite SMILES
find_met_smiles
kcats_from_brenda
kcats_from_ec_tree
kcats_from_homologs
kcats_from_dlkcat
kcats_from_okp
data/kcats/custom.tsv
**Applying kcats**
assign_kcats (preview: collect_kcats)
fill_kcats_from_isozymes,
assign_standard_kcat (optional)
apply_kcat_constraints
set_prot_pool_size
model.optimize: growth reaches the target?
yes
no
no, several conditions measured
**Growth-rate tuning**
sensitivity_tuning
fit_sigma
**Tuning against experimental data**
screen_kcat_leverage
review_assignment
cmaes_kcat_tuning
Functional ecModel
save_ec_model
**Proteomics integration**
load_prot_data,
fill_enz_concs
level_complex_concs,
constrain_enz_concs
calculate_f_factor,
set_prot_pool_size
load_flux_data,
apply_flux_data_constraints
Growth reaches the target?
yes
no, gradual
no, fast
no, smallest total amount
without proteomics
**Relaxing**, gradual
flexibilize_enz_concs
**Relaxing**, fast
relax_proteomics_greedy
**Relaxing**, smallest total amount
relax_enz_concs
ecModel for simulation and analysis
**Simulation**
ec_fva,
random_sampling
ec_fseof
get_subset_ec_model
**Enzyme usage**
enzyme_usage
get_enzyme_bottlenecks