```
# Vary the physical and biogeochemical configuration across the N ensemble members.
# We apply log-normally distributed scale factors to:
# * wind speeds (x and y components)
# * background mixing (minimum turbulent kinetic energy)
# * maximum growth rates of the two phytoplankton types
# Each member will have different configuration files for physics (gotm.yaml)
# and biogeochemistry (fabm.yaml). The path to the latter is set in the former.
# Configuration files for each member are written when the "with" clause exits.
rng = np.random.default_rng()
gotm = eatpy.models.gotm.YAMLEnsemble("gotm.yaml", N)
fabm = eatpy.models.gotm.YAMLEnsemble("fabm.yaml", N)
with gotm, fabm:
    gotm["surface/u10/scale_factor"] = rng.lognormal(mean=0.0, sigma=0.2, size=N)
    gotm["surface/v10/scale_factor"] = rng.lognormal(mean=0.0, sigma=0.2, size=N)
    gotm["turbulence/turb_param/k_min"] *= rng.lognormal(mean=0.0, sigma=0.2, size=N)
    gotm["fabm/yaml_file"] = fabm.file_paths
    fabm["instances/phy/parameters/mumax0"] *= rng.lognormal(mean=0.0, sigma=0.2, size=N)
    fabm["instances/dia/parameters/mumax0"] *= rng.lognormal(mean=0.0, sigma=0.2, size=N)

# Vary the initial state across the N ensemble members, using log-normally distributed
# scale factors drawn for each member and variable.
# Restart files for each member are written when the "with" clause exits.
restart = eatpy.models.gotm.RestartEnsemble("restart.nc", N)
with restart:
    for name, values in restart.template.items():
        shape = (N,) + (1,) * values.ndim
        scale_factor = rng.lognormal(mean=0.0, sigma=0.2, size=shape)
        restart[name] = values * scale_factor
```