build.py -e TEST -v 0|last -f mytrain.sdf -m imodel.py

-e TEST Endpoint label. We recommend using a short label in uppercase.

-f mytrain.sdf Name of a SDFile with all the compounds in the training series. Must contain the activity in numeric format, in a field labeled as <activity>.

-m imodel.py Name of a model definition file in Python, based on the provided templates.

-v 0|last Missing elements required for building the model will be taken from corresponding version folder. The version is provided as a number or as the text "last" to make reference to the last model version.

This command is used to build a new predictive model, using the training set and the model definition file defined by the arguments, as described above.

New models are always created into the "version0000" directory, overwriting previously unpublished models for this endpoint. The user can refine the model and once satisfied the model can be published using the command "*manage -e TEST --publish"*

New versions are added incrementally as version 1, 2, …

When the command contains the name of a training series or the specified model definition changes the normalization of previous models, this command will run the steps of the workflow required to regenerate the matrix of descriptors. Otherwise, the command only rebuilds the mathematical model.

Examples of use:

build -e TEST -f new.sdf -m imodel.py

This will build a new model using the provided training (new.sdf) and model (imodel.py) files.

build -e TEST -v last -f new.sdf

This will build a new model using the imodel.py from last version.

build -e TEST -v 2 -f new.sdf

This will build a model using new.sdf as the training series and the imodel.py obtained from version0002 directory

build -e TEST -v last -m imodel.py

This builds a new model using the training.sdf from last version

predict.py -e TEST -v 0|last -f mols.sdf

-e TEST Endpoint label

-**v 0|last** Number of the model version used for the prediction or the label "last" for indicating the last version

-f mols.sdf Name of a SDFile containing one or more molecules

This command is used to carry out a prediction on the molecules provided as argument using the specified model.

Examples of use:

predict -e TEST -v last test.sdf

This predicts the modeled endpoint for all compounds in test.sdf using last model.

manage.py -e TEST -v 0|last --publish|new|remove|version|info=[short|long] |get=[model|series][-t tag]

This command performs diverse eTAM administrative tasks, including creating new endpoint trees, publishing model versions and showing model information. It is strongly recommended that all the eTAM model manipulation is carried out using this command.

-e TEST Endpoint label. This argument is compulsory for "publish", "new", "remove" and "get", and optional for "info".

-v 0|last Number of the model version used or "last" to indicate the last model version. This argument is compulsory for "get" and optional for "info".

-- publish Creates a new version starting from the model present in the "version0000" folder of a given endpoint. This procedure copies all the contents of the folder, including the training set and the model definition files. The number of the new version is created incrementally over previous versions (e.g. if the last version is the 2, this procedure will create version 3)

--new Creates a new endpoint (e.g. TEST in the example), with an empty "version0000" folder. This command requires the "-e" argument for assigning the endpoint name as well as the "-t" argument for defining the label of this endpoint in the eTOXsys web service

--remove Removes the last model version for a given endpoint. All the contents of the folder and the folder itself will be removed. The version 0 cannot be removed with this command. The actions of this command cannot be undone, please use with extreme care.

--version Shows the current eTAM version

--info=short|long Presents information about the model or models defined by the endpoint and version arguments, using the specified format. If no endpoint is specified, the command shows models for all the endpoints. If no version is specified, the command shows all the model versions.

--get=model|series Retrieves the required information from the model defined by the endpoint and version arguments. Both of these are required. If used with the "model" argument, the command copies the model definition file "imodel.py" to the current directory. If used with the "series" argument copies the training series "training.sdf" to the current directory. Please notice that this last file is a verbatim copy of the original file used to build the model, before the normalization.