831472 - MELLODDY

MachinE Learning Ledger Orchestration for Drug DiscoverY

WP1 – Pre-processing of data up to a level of necessary and sufficient standardization

D1.3 Data Preparation Manual for Year 3

Lead contributor	Ansgar Schuffenhauer (13 – Novartis)
	ansgar.schuffenhauer@novartis.com
Other contributors	Adam Zalewski (16 - Amgen Research Munich GmbH)
	Hideyoshi Fuji, Arina Afanasyeva (14 - Astellas)
	Ola Engkvist, Mervin Lewis (8 - AstraZeneca AB) (8 - AstraZeneca AB)
	Andreas Goeller, Tobias Morawietz, Anastasia Pentina (12 - Bayer AG)
	Bernd Beck, Lina Humbeck (11 - BII GmbH)
	David Marcus, Stephen Picket (9 - GSK)
	Nicolas Do Huu, Brice Hoffman, Nawfal Tachfine, Nicolas Drizzard, Rama Jabal (5 - Iktos)
	Wouter Heyndrickx, Thanh Le van (7 - Janssen)
	Yves Moreau, Martijn Oldenhof (6 - KU Leuven)
	Michael Krug, Lukas Friedrich (10 - Merck KGaA)
	Nikolas Fechner, Noe Sturm (13 - Novartis)
	Arnaud Gohier, Peter Schmidtke (15 - Institut de recherches Servier)

Due date	31 October 2021

Delivery date	29 October 2021
Deliverable type	R
Dissemination level	PU

Description of Work	Version	Date
	V1.0	25. October 2021

Document History

Version	Date	Description
V0.1	17. Apr 2021	First Draft
V0.2	1.Oct 2021	Second Draft
V0.3	5. Oct 2021	Third draft, addresses catalog task generation and data set hand-
V0.4	18. Oct 2021	4th draft, added auxiliary pseudo label data processing, memory profiling
V0.5	22. Oct 2021	Added the guidance for qualification of auxiliary data (use validation set)
V1.0	25 Oct. 2021	Final version submitted to WP7

Publishable Summary

This deliverable is produced for the purpose of documenting and reporting the progress that has been made within the MELLODDY project. MELLODDY will demonstrate how the pharmaceutical industry can better leverage its data assets to virtualize the drug discovery through the development of a secure and privacy-preserving platform for federated machine learning. The timings and condition of this deliverable was defined by the Annex 1 of the MELLODDY Grant Agreement N° 813472.

This deliverable describes the data preparation for the private pharma data as well as the included public data, which each partner must execute on its own data according to the common and agreed procedure. This procedure ensures that the data of all partners will be presented in a consistent way across all partners. Only the output of this preparation process will be exposed as training data to the federated machine learning process.

Abbreviations and definitions

ECFP	Extended Connectivity Fingerprint
LSH	Locality Sensitive Hashing
ML	Machine Learning
ADME	Absorption, Distribution, Metabolism, Excretion
QC	Quality Control
AUC-ROC	Area Under Curve-Receiver operating characteristic
Trunk model	The part of the neural network that is shared across the pharma partners
Individual Code	data preprocessing code to generate input for the common code written by each partner individually
Common code	Part of data pre-processing code that is written by Merck KGaA and used by all pharma partners
MELLODDY-TUNER	MELLODDY Tool for UNifying and Encrypting of data Records, also referred to as common code
HTS	High-Throughput screening
prediction task	A column in the Y matrix. This is linked to an assay. For classification with multiple thresholds more than one classification task per assay may exist
AUCPR	Area under the precision recall curve
CRO	Contract research organization
Catalogue assay	Assay protocol publicly offered for execution by a CRO and identifiable by a catalogues ID

Table of content

1 Introduction

The purpose of this manual is to ensure that all data contributing partners prepare the input structure activity data for the federated machine learning according to the same standards and principles. This is required to ensure a consistent representation of chemical structures and the biological activity data. The steps described in this manual must be executed by each company individually on its own compute platform. Only the output of this preparation process is made accessible to the federated machine learning process. This manual describes the data preparation for the third federated machine learning run, which is planned at the end of the third project year.

In this manual example data from ChEMBL release 25 (10.6019/CHEMBL.database.2510.6019/CHEMBL.database.25) is used to illustrate the workflow and to describe the data at the various processing stages) and cover the assay IDs:

- CHEMBL3855277
- CHEMBL3855278
- CHEMBL3855279
- CHEMBL3855298

Individual data points have been modified, in order to be able to illustrate the preparation process. On a second note these assays have data for some enantiomer pairs, which have been separated, but where the absolute stereochemistry has not been assigned. As a consequence several distinct samples were linked to the same ChEMBL ID. In addition, in order to comply with the requirement for the common data preparation code (see below) all input ID from ChEMBL have been stripped of the “CHEMBL” prefix to obtain IDs in integer format.

In addition, a purely fictitious single concentration HTS assay with the ID 9999999 has been added for illustration purposes.

An overview of the data preparation process can be found in Figure 1: General data preparation workflow. The labels T0 to T11 in this figure refer to example data tables. . The data preparation is divided in two major stages, where the first one involves exporting the data from the individual data warehouses of the partners. This step will involve replicate aggregation at the sample level. This will be done with code individually created by the data contributing partners in order to accommodate their individual data warehousing systems (“individual code”). In some cases, only guiding principles have been agreed upon, leaving the details of implementation to the individual partner.

The second stage involves structure processing, descriptor calculation, fold assignment and replicate aggregation. This will be performed by a common python script provided by Merck KGaA in collaboration with other partners according to the specifications in this manual. (“common code” also referred to as “MELLODDY-tuner”, D1.5 and D1.8).

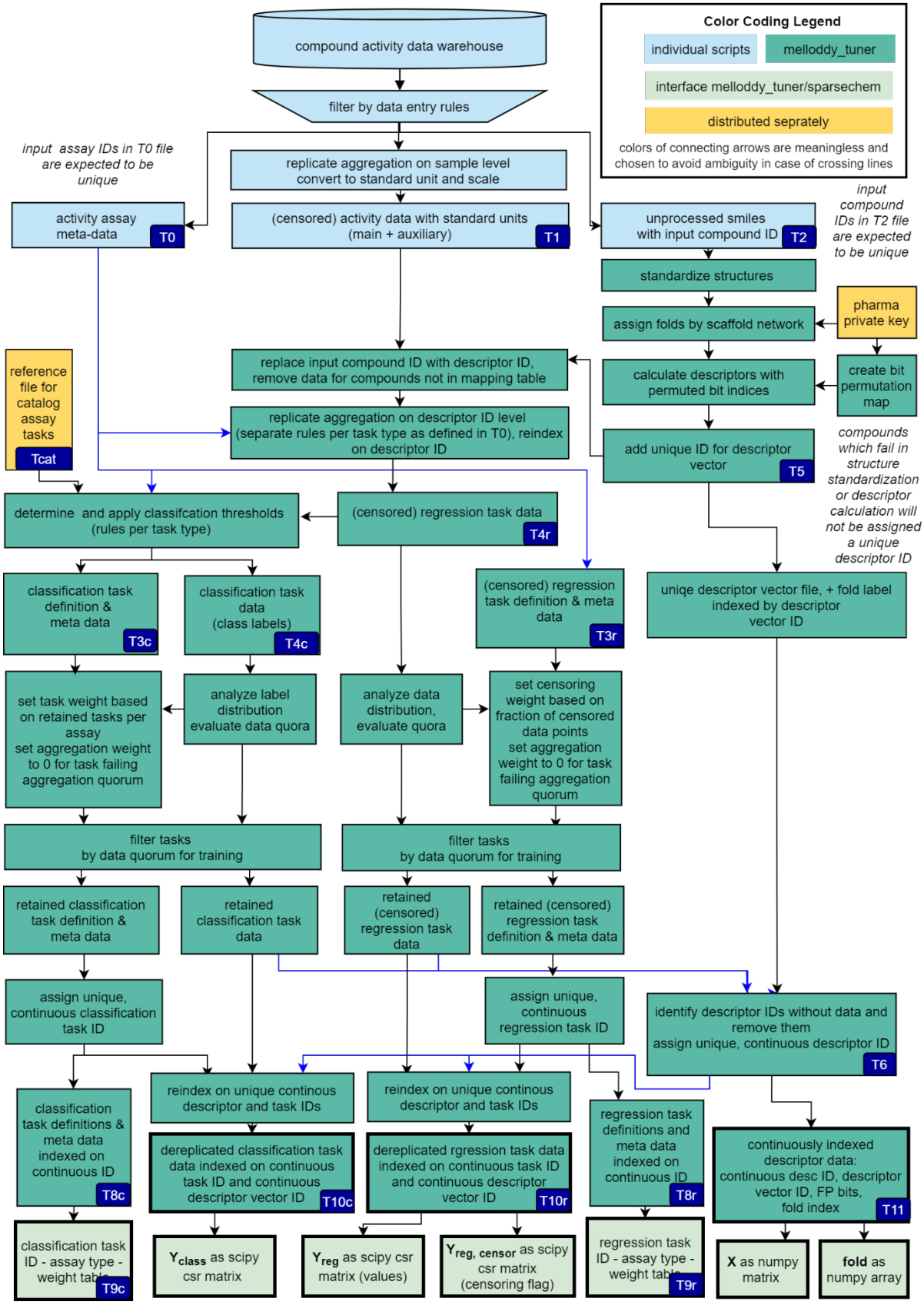

Figure 1: General data preparation workflow. The labels T0 to T11 in this figure refer to example data tables. Due to the changed workflow with respect to the year 1 data preparation, the numbering of the tables no longer follows the sequence in which they are generated, however, for consistency purpose the nomenclature used in year 1 was kept whenever possible.

Table	Description	Usage
T0	Assay metadata. Additional data can be carried through until the last step for training.	Input for MELLODDY-tuner provided by pharma as part of their individual code
T1	Main prediction endpoint data as (censored) quantitative values with normalized units, an assay identifier referencing T0 and a sample identifier referencing T2	
T2	File containing the chemical structures for the samples as smiles	
Tcat	Reference file used to assign a unique catalogue task_ID for each combination of catalogue_assay_ID and threshold	Assign of unique catalog task Ids across partners
T3c	Metadata for classification tasks, including classification thresholds	Intermediate files and tables created by MELLODDY-tuner
T3r	Metadata for regression tasks	
T4c	Data table classification task data after replicate descriptor vector aggregation	
T4r	Data table with regression task data after replicate descriptor vector aggregation.	
T5	Mapping table input compound ID – unique descriptor vector ID	
T6	Descriptor data file having one row per unique descriptor vector ID	
T8c	Metadata for classification tasks, including classification thresholds after removal of entries not fulfilling data volume quorum and with renumbered continuous classification task_id Pharma partners will use this table to map back their model predictions to their original assays	
T8r	Metadata for regression tasks, after removal of entries not fulfilling data volume quorum and with renumbered continuous regression task_id. Pharma partners will use this table to map back their model predictions to their original assays	
T9c	Same as T8c, but with all metadata not required for training removed, including input_assay_id	Data files for machine learning in human readable text format before activity formatting
T9r	Same as T8r, but with all metadata not required for training removed, including input_assay_id	
T10c	Data table with classification task data after replicate descriptor vector aggregation	
T10r	Data table with regression task data after replicate descriptor vector aggregation	
T11	Compound descriptor data with associated fold index	

$\mathbf{Y_{class}}$	Sparse scipy csr matrix for classification task data
$\mathbf{Y_{reg}}$	Sparse scipy csr matrix for regression task data
$\mathbf{Y_{censoring}}$	Sparse scipy csr matrix holding censoring for for $\mathbf{Y_{reg}}$
X	Compound descriptor data a sparse numpy matrix
folds	Fold assignment vector

Table 1: Overview of tables during data processing

2 Data Entry Criteria

2.1 Included Assays

2.1.1 Primary Prediction Endpoints

Concentration-response bioactivity assays and assays for physical-chemical or ADME related properties are considered as the primary prediction endpoints, for which the model performance should be optimized. Concentration response assays will give as main readout a value called IC_{50}, EC_{50}, AC_{50}, K_i, K_D or similar in a concentration unit, resulting from a fit of a concentration-response curve to the Hill equation (sigmoidal curve). These are the values to be included. The additional fit parameters from the hill equation (Hill slope, A_{inf} and A_0) are not included as model endpoints. They may be used to guide the decision whether to include a data point, if such filtering has not happened already at the stage of curve fitting and data warehouse entry.

Background

In a concentration-response assay (also called dose-response assay) a dilution series is produced for each compound with concentrations regularly spaced on the logarithmic scale reaching for example from 30 µM to 1 nM concentration. For each concentration in this dilution series the assay signal is read out, and the normalized assay signal as function of the logarithmic concentration is (typically) fitted as a sigmoidal concentration response curve. The inflection point of this sigmoidal curve is used as the main numerical value summarizing such a dose response experiment and is called IC_{50}, AC_{50}, EC_{50} or similar, depending on the type of activity. If the inflection point lies within the concentration range covered by the dilution series the values are typically reported as exact values (e.g. IC_{50} =0.5 µM). For cases in which the inflection point is lying outside the maximum measured concentration (extrapolated inflection points) or the concentration response curve is flat, qualified values are usually reported (e.g. IC_{50} >30 µm). For some instances, qualified values can occur within the concentration range, if, for example, some measurements of the dilution series become unusable because of confounding effects at higher concentrations such as cytotoxicity and therefore have been excluded from curve fitting. For highly potent compounds, extrapolated inflection points can lie below the lower limit of the dilution series and qualified values may be reported (e.g. IC_{50} <0.001 µM).

Types and units of ADME-related assays are typically more variable, and guidance for inclusion is given below in §3.2.3

In general, each assay is treated as its own prediction endpoint. This means that there is no aggregation done based on the assay target (*with the exception of public data, as described below*). The rationale for this is the following:

- Different assays on the same targets are not expected to give the exact same numerical values, since many different assay principles are used which can lead to weak correlation between the assays. Even in case of the same or similar assay principle, simple dose response read-outs such as IC_{50} values are depending on assay conditions such as the concentration of reagents or target protein. Since the partners are interested in obtaining (semi)quantitative models for at least a part of their assays, this requires the numeric outputs to be comparable.
- The inclusion of assays not tightly coupled to an individual target biomolecule is possible and encouraged.

The assay identity is thereby established by the assay registration systems used at the individual partners. It is explicitly acknowledged, that the identity criteria may vary from partner to partner. For example, when a concentration response assay is handed over from one lab that established and used it for validation of results from a single concentration HTS campaign to another biology lab running it for the purpose of further lead optimization, some partners may choose to treat both assays as different, whereas other partners still treat this as an identical assay.

In most cases relying on the partners' own assay registration system to establish assay identity is sufficient, as assays are typically not shared among the partners. There is however an exception to this for the "catalogue assays", which are offered by CROs on a fee-for-service basis and are used by multiple pharma

companies. In this case, multiple companies may actually own data from an identical assay. The partners using such catalogues assays may therefore wish to combine their data into joint prediction endpoints. In order to enable this, partners can map their internal assay ID in the input to published catalogue assay IDs. Catalogue fusion is only applied to classification tasks, but the corresponding tasks can be still used in regression without catalogue fusion since under the standard paradigm each partner's assay is a separate task.

In the public data there is no assay registration at all, and assay identity is typically tied to the underlying scientific publication or patent.

In some cases one assay may produce more than one relevant read-out, as for example a solubility assay may report solubility values at multiple pH values. In such a situation each relevant read-out should be treated as its own prediction endpoint and have its unique input_assay_id.

2.1.2 Auxiliary data

Also in the third project year, in addition to the dose-response assays as the primary prediction endpoints, we will include auxiliary data. Auxiliary data will be used in the training to fit the model, but will not be included in model performance evaluation.

The following auxiliary data categories are considered for year 3:

Single concentration high-throughput screening data: This data will typically result from plate-based screens. The raw read-outs are expected to be normalized by on-plate controls. Typically, a neutral (low) control and an active (high) control are used, in order to create a percent activity read-out. However, also normalizations only to a low control are acceptable. We expect HTS data to be quality controlled and normalized. Confounding effects such as edge effects or liquid handling artefacts have either been corrected for or the affected data points been removed.

Pseudolabel data: This type of auxiliary data results from high content experiments such as cellular imaging or transcriptomics. Imaging data will typically result from plate-based imaging screens, where images are acquired by an automated microscope and are then processed automatically by an image analytics software such as Acapella or CellProfiler. This will involve typically an expert designed segmentation process to detect cellular components and taking measurements on them. In either case a dense matrix of experimental features for each sample is generated in such experiments. These features are not used directly as auxiliary prediction tasks, but are used to generate so-called pseudolabels, which are predictions of main endpoints based on the auxiliary data.

Generation of auxiliary pseudo labels is two-step process. The first step requires a dense feature matrix X_{aux} resulting from a high content experiment such as imaging as input alongside with the main task Y matrix, in order to train a model predicting main task values from the feature matrix. In the second step, this model is used to predict the main task Y data. The resulting predictions are filtered by quality, retaining only high quality predictions. This results in a data matrix Y_{aux} of predicted pseudo labels, which is much denser than the original Y matrix, but contains only a subset of tasks amenable to reliable prediction. This Y_{aux} matrix is then combined with the original Y matrix (see also Figure 2). Y_{aux} from pseudolabels is directly generated in binary form.

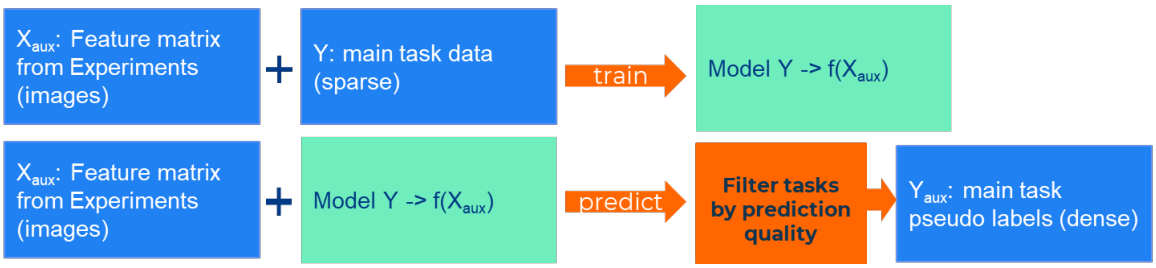

Figure 2: Generation of auxiliary pseudo-label tasks

In case there is more than one feature matrix, this process must be repeated for each feature matrix. The task quality filter criterion is a parameter, which each partner will have to optimize by himself. The use of such auxiliary data per partner is optional and not expected from every partner.

Both types of auxiliary data will be brought in as classification tasks only. It is however possible in a hybrid model setup to include both regression and classification tasks, that also regression tasks can potentially benefit from auxiliary data.

Given that auxiliary data is introduced for the exclusive purpose to improve the prediction of the main endpoints, partners are asked to assert individually for their auxiliary data that this is actually the case. An auxiliary data set may only be introduced if the following conditions are met in a single partner setup.

1. The aggregated primary performance metric across all endpoints must not suffer from the introduction of auxiliary data

2. At least one of the following criteria must be met:

    a. The primary performance metric of a hyper-parameter optimal model with auxiliary is larger than the corresponding metric of a hyper-parameter optimal model without auxiliary data.

    b. The domain of applicability was extended as measured by conformal predictor efficiency increases when introducing auxiliary data

3. Main endpoint tasks closely related to auxiliary tasks must benefit. For single concentration HTS assays, these are the main tasks corresponding to the concentration response validation of the HTS data. It is understood that it may not be possible to identify a pairing for each HTS assay. The benefits of HTS data should assessed on that subset of main endpoints that can be linked to a HTS single concentration assay. For pseudolabel tasks, the main task from which the pseudolabels were derived is always known and needs to benefit. Benefit needs to be established either as:

    a. Improvement of primary performance metric

    b. Increase of applicability domain

These criteria above should be evaluated by training models on the three training folds and evaluate the performance on the validation fold. The independent test fold should **NOT** be used.

2.1.3 Regression tasks

In the year 3 run partners are requested to include only such task into the regression model, that have the chance to result in a meaningful model. As a guidance the following criteria are provided:

- Tasks with categorical, discrete values are not suited for regression

- Tasks need to have a minimal level of variance. The standard deviation across all observations needs to exceed the standard error that is expected from repeated measurements on the same sample. For pIC_{50} type assay readouts, this experimental error can be assumed to be at least 0.3 log-units.

- Tasks for which partners were successful in building an in-house regression model are generally acceptable.

- If partners wish to evaluate whether for a task a reasonable model can be built, they should do so by training on the three training folds and then evaluate the task performance on the validation fold

The fact that a task is considered not suitable for regression does NOT imply that it cannot be used as a classification task, where the aim is still to be very inclusive (see also 3.8.1).

Non-ADME Tasks that have a standard deviation below 0.5 log-units in one or more folds are automatically de-selected for evaluation (aggregation_weight is set to 0.0). These tasks are still used in training if the training quorum is met.

2.2 Chemistry Space Criteria

Small, organic molecules, without any exclusion criteria by chemical attractiveness, drug-likeness or similar empiric rules, represent the chemical space of the model training.

In order to prevent the structure processing described below to take excessively long for individual extremely large molecules the number of non-H atoms is limited to ≤ 100. This limit is enforced by the common structure processing code (see §4.1) to ensure that the restriction to small molecule data is fulfilled.

2.3 Data Volume Criteria

In order to ensure that enough data are present for training and reliable calculation of performance figures, two types of data volume criteria are used

- **Training quorum**: This quorum must be met or exceeded so that the task / assay can be included in the training set at all. This quorum will be checked by the MELLODDY-tuner code after replicate aggregation. However, as replicate aggregation will possibly reduce the number of data points it would be good practice to filter accordingly before processing it with MELLODDY-Tuner, in order to speed up the processing.

- **Evaluation quorum**: This quorum must be met or exceeded so that a task is included in the calculation of the global performance metrics. It must be at least as stringent as the Training quorum, in order to take effect, as tasks not included in the training cannot be included in performance metric calculation anyway. This quorum is meant to ensure that only such tasks will be included in the aggregated performance figure calculation, that have enough data points in the validation or test fold to ensure that performance figures can be calculated accurately.

The values for these quorums are dependent on model type (classification versus regression and also on the assay type).

Assay type	Quorum Type	Classification	Regression
Primary Prediction endpoints NON-CATALOG-PANEL OTHER ADME	Training	• 25 observations per class label (active/inactive) in whole dataset	• 50 observations in whole dataset • 25 observations without qualifier in whole dataset
	Evaluation	• 25 observations per class label (active/inactive) in each fold	• 50 observations in each fold • 25 observations without qualifier in each fold • For non-ADME: Standard deviation > 0.5 in all folds
Primary Prediction endpoints CATALOG-PANEL	Training	• 400 observations in total	Not applicable (catalogue fusion is not used for regression) CATALOG-PANEL tasks are treated as NON-CATALOG-PANEL tasks.
	Evaluation	• 25 observations per class label (active/inactive) in each fold	Not applicable (catalogue fusion is not used for regression) CATALOG-PANEL tasks are treated as NON-CATALOG-PANEL tasks.
AUX_HTS AUX_PL	Training	• 10,000 measurements • 10 actives	Not applicable (no auxiliary data in regression for year 3)
	Evaluation	Not applicable	Not applicable

Table 2: Minimal data volume quorum for training and evaluation by assay type

2.4 Data Quality

Experimental results are obtained from a physical sample, to which a chemical structure has been attributed, in an assay that serves as reduced model system for biological or biochemical process, such as for example binding to a specific target. From these data we wish to conclude that a molecule with attributed structure causes the biological effect the assay intends to measure. In reality, the samples are almost never 100% pure, but contain impurities, and likewise most assays are vulnerable to some forms of perturbation, which will result in a positive assay outcome even in absence of the biological affect the assay was designed for.

2.4.1 Sample Quality

In order to ensure that the assay results in the data set can be truly attributed to the nominal structure, ideally the identity and purity of the sample should be verified at the time of experiment. While this goal cannot be fulfilled in practice, all pharma partners agreed to adhere to their internal good practice guidelines and will avoid including data generated on impure samples or samples with an incorrectly attributed structure to the best of their abilities.

Background

In industrial compound collections, analytical compound sample quality control (QC) is performed in at least one of the two stages below to ensure chemical integrity of the sample:

Initial sample QC by the synthesizing chemist. Though typically there apply company guidelines for minimum purity requirements, the analytical methods deemed appropriate may vary for the compound structure in question. Low throughput methods are acceptable, as the analytical measurements and their interpretations are distributed over the chemists.

Analytical QC of the screening stock solutions at the time the screening stock solution was produced or the assay was run. This needs to run in high-throughput and is typically done by liquid chromatography coupled to mass spectrometry and UV readout (LC/MS). The UV trace is used to establish the sample purity, whereas the MS is used to confirm whether the molecular mass under the main peak is in agreement with the expected mass of the nominal structure. The measurements and interpretations in this case will be highly standardized and automated to cope with the necessary throughput.

In either scenario the purity information may be available explicitly in numerical form, or only implicitly ("if it has a registration number, it is at least X % pure"). Implicit purity criteria make it impossible to require a fixed level of purity across all partners. Older sample may not have undergone the purity check of current industry standards. In any case sample QC is a snapshot in time and does not guarantee that the sample has fulfilled the purity requirement at the time the assay data point has been generated.

2.4.2 Frequent Hitters

The treatment of frequent hitters is left to the individual partners, based on necessity. In general we expect that frequent hitters will be removed from the auxiliary HTS data.

Background

While most of the assays have been designed and optimized to measure the interaction of small molecules with a single target, protein complex or pathway, in practice most assays are sensitive to a at least some unspecific interference mechanisms such as aggregation[1], redox cycling[2], metal contamination effects[3], electron transport chain inhibitors[4], in order to name only a few examples. While there is the expectation that these specific mechanisms can be addressed with appropriate counter assays, in practice it is not always straightforward to identify the mechanism that makes a specific compound or compound class a frequent hitter. The analysis of the complete screening history of the compound as for example described by Beck[5] is thus often a more efficient and pragmatic approach to identify such compounds. Sometimes it is possible to define substructure patterns from such empirical frequent hitter data, as for example the PAINS filters.[6] While typically only a small minority of a screening collection shows a frequent-hitter behaviour, depending on the robustness of the assay, a significant fraction of the active compounds may be frequent-hitters.

The impact of frequent hitters on ML model building is still poorly understood. In a setup where the primary modelling endpoints are assays and not targets (see §2.1.1), the prediction of a frequent hitter as a compound active in the assay is formally a correct prediction. Still, the concern is that such frequent hitters could take up an unduly large fraction of model degrees of freedom or could inflate the performance metrics, as predicting mostly frequent hitters as active might be a more straightforward and trivial task than modelling truly active compounds.

Single partner studies showed that the impact of frequent hitters on the modelling results differs between the partners. This is a consequence of the different practice between pharma partners with respect to the stage at

which frequent hitters are removed. For partners which remove frequent hitters after single concentration HTS before selecting compounds for follow-up in concentration response assays have often less frequent hitters in their dataset compared to companies applying such a setup only after progression of compounds to concentration-response assays.

3 Individual Data Processing

3.1 Overview of tasks to perform

The partners will export their structure activity data from their data warehouses. During this export the following tasks must be performed:

- Assignment of assay types (§3.2)
- Assignment of catalogue assay IDs (§3.2.3)
- Unit harmonization and scaling (§3.3)
- Optional definition of expert thresholds (§3.4)
- Replicate aggregation on sample level (§3.6)

In addition pharma partners can take the necessary steps that the year 3 MELLODDY Tuner input is also suitable for time gated analysis of the year 2 models. These steps are optional. They are described in §3.7.

The result of this export is a set of three files as described in §3.8:

1. **T0**: A unique list of assays. This file has to contain a unique input_assay_id, alongside with required metadata such as the assay type (§3.2) and optional expert thresholds (see §3.4)
2. **T1**: A table of assay results representing primary prediction endpoints. This table needs to reference **input_assay_id** from T0 and **input_compound_ID** from **T2**. It contains the activity data after unit harmonization, scaling (§3.2.3) and sample replicate aggregation (§3.6).
3. **T2**: A table of chemical structures associated to the **input_compound_ID. input_compound_ID** in this table is expected to be unique, whereas replicate smiles are acceptable and expected.

3.2 Assignment of Assay Types

Background

Besides project specific assays, pharma partners typically run assays which are used by many different projects to characterize their compounds with respect to physical-chemical and other ADME (Absorption, Distribution, Metabolism, Excretion) properties as well as biological assay panels to discover undesired off-target interactions related to adverse effects ("safety" panels). Because of their use across individual projects, these assays are exposed to more diverse chemical matter than assays used in the context of an individual discovery project, which should make them more amenable to machine learning. Also, the endpoints of these assays are overlapping to a large degree between the partners, indicating higher potential to benefit from the federated approach. Thus, if the federated approach is not able to show superiority across all assays in the second year, we may at least be able to demonstrate superiority for such assay types which is expected to be less challenging to achieve.

The pharma partners will group their assays and the resulting classification and regression tasks into assay types according to their role in the drug discovery process. This type of annotation is based on the general operation mode for drug discovery in the pharmaceutical industry and is not informative in any way about the

partner's disease, target or assay technology portfolios. The classification scheme is described below in Table 3: Assay types.

Assay types are used for two purposes:

- During the automated data preparation in MELLODDY-Tuner, some steps of the processes are done differently for different assay types, such as for example automated threshold setting to define classification tasks.

- During performance evaluation the results for the different assay types will be also analysed separately.

Assay Type	Characteristics	Chance to observe improvement in multi-partner ML
ADME	• Used to optimize the pharmacokinetic properties • Assays run for a long time • Diverse compounds from a wide range of projects • High overlap of endpoints between pharma companies • Assays may be identical between partners because some contract these out to service providers • Readout often not dependent on small molecule – target interactions	High success chance based on y2 results
NON-CATALOG-PANEL	• Used to optimize selectivity against off-targets with the aim to anticipate adverse events • Assays run for a long time • Diverse compounds from a wide range of projects • High overlap of endpoints between pharma companies (see Bowes *et al.* [7]) • Assays may be identical between partners because some contract these out to service providers • Often compounds are submitted to a whole panel at a time, making the part of the activity matrix less sparse, but by far not fully filled, as panels change over time and are often also structured into subpanels.	Increased success chance over OTHER expected
CATALOG-PANEL	• Same as NON-CATALOG-PANEL, but used in catalog fusion	High success chance
OTHER	• Typically used to identify molecules with on target activity or to optimize their potency • Overlap of endpoints between pharma is lower than in the other two types • Especially in case of assays used in lead optimization stage of an individual project, the number of compound classes having data in this assay will be limited • Vast majority of assays will fall in this type	More Challenging
AUX_HTS	Auxiliary task from single concentration HTS, contribution only to training, but not to evaluation metrics	Not applicable
AUX_PL	Auxiliary data generated as pseudo-labels, contribution only to training, but not to evaluation metrics	Not applicable

Table 3: Assay types

3.2.1 Which assays should go into the category ADME?

One key criterion is that the assay has been or is used across a wide range of projects. Examples of assays to go into ADME category are:

- Solubility assays

- logP and logD@pH
- pKa, fraction ionized
- Permeability (MDCK, Caco-2, PAMPA etc.)
- Membrane affinity
- HSA binding, Plasma protein binding
- Liver microsomal or hepatocyte stability

3.2.2 Which assays should go into the category NON-CATALOG-PANEL?

One key criterion is that the assay has been or is used across a wide range of projects. This means project specific selectivity assays, such as for example an assay addressing sub-type selectivity for the project's target should not go into the PANEL category. Examples of PANEL assays are:

- general pharmacology safety panels
- Ion channel assays addressing cardiac safety (e.g. hERG)
- target family specific selectivity panels, such as kinase panels
- CYP inhibition panels

Phenotypic toxicity assay such as Ames tests or phosholipidosis assay should not be assigned to the PANEL type, but rather be treated as OTHER.

3.2.3 Which assays should go into the category CATALOG-PANEL?

They key criterion for putting an assay into this category is that the assay was run as catalogue assay at a CRO, to which multiple customers can submit samples for screening. Assays in this category must be mapped to a reference catalogue of assays offered by CROs, which is made available to the pharma partners. In order to map an internal assay to a given catalogue assay, the pharma partner must be certain without any doubt that the data resulted from that catalogue assay. In addition, the specific data volume quorum as set out in Table 2 must be met.

3.3 Unit Harmonization and Scaling

3.3.1 Scaling of Concentration Response Data

Concentration response activity will give read-outs like IC_{50}, AC_{50}, EC_{50}, K_i, K_D, which are in molar concentration units such as µM. The original values in molar are transformed into the negative logarithm to the basis 10 to give pIC_{50}, pAC_{50}, pEC_{50}, pK_i, pK_D. Some conversion examples are shown below:

Input activity	standard_qualifier	standard_value
0.1 mM	=	4.0
> 30 µM	<	4.5
0.1 µM	=	7.0
< 0.001 µM	>	9.0
10 nM	=	8.0

Please note that due to the use of the negative logarithm also qualifiers will have to be inverted.

The resulting standard_values are expected to be in the credibility range from 3-10.

3.3.2 Use of single concentration data for CRO profiling assays

Normally single concentration data should not be used as primary endpoints. An exception is made for CRO based profiling panel assays, which will often be catalogue assays. In many such assays the CRO starts with a single concentration screen, and only follows up with a concentration response measurement, if the activity in the single concentration measurement is sufficiently high (typically 50%). In order to make use also of the single concentration measurements, and thus increase the number of measurements, the following approach is suggested:

- If a concentration response curve is measured, the data from this is used.
- If single concentration data exist, and no single concentration data point has an activity greater than 25%, then this will be treated as inactive example, with an IC_{50} > [maximal single concentration measured]

Reported IC_{50}	%activity @ 10 uM	%activity @ 30 uM	pIC_{50} used for T1 file preparation
4.6 uM	72	96	5.3
> 30 uM	16	12	< 4.5
> 30 uM	20	35	< 4.5
	3		< 5
		3	< 4.5
	41		Don't use this data point
		41	Don't use this data point

Table 4: Example for catalogue CRO assay preparation. This example is built on IC_{50}, but applies in analogy also to AC_{50}, EC_{50} and other concentration-response summary values

This representation is mandatory for all catalogue assays undergoing catalogues assay fusion (type CATALOG-PANEL), but also recommended for other CRO panel assays run under the same operational model.

3.3.3 Scaling of ADME Assays

The read-outs of ADME assays are more heterogeneous than those for bioactivity assays. For common types of assays the rules have been specified in Table 5. Pharma partners can, over the course of MELLODDY, mutually agree to define rules for additional types of ADME assays. The types of ADME assays that can be included are not limited to those assays for which common scaling rules have been agreed upon. As a general rule for those assays not listed in the table, the distribution of values should be analyzed. If the data are rather lognormal distributed than normal, a log-transform is suggested.

In addition, this table also provides a direction of interest. This will be used in automated thresholding to decide which direction of the scale should get the "active" label. As some performance metrics like AUCPR are not symmetric, it is important to define to which end of the scale precision and recall should refer to. In general, the direction chosen here is the end of the scale, at which ADME problems and obstacles are to be expected. However, not for all ADME assays this direction make sense, and for such assays asymmetric metrics like AUCPR are most likely not suitable. The direction of interest is applied to the unit transformed and scaled data. During the transformation, the directionality may change, as for example in the case of PLASMA Protein binding, where a high plasma protein binding leads to a low value of logKa.

Assay	**Measurement**	**Standardized Unit and Scaling**	**Direction of interest**
Permeability assays (caco-2, MDCK, PAMPA)	Permeation coefficients P_{A-B}	10^{-6}cm/s, Log10 scaled	low
	Efflux Ratio (=P_{B-A}/P_{A-B})	Unitless, Log10 scaled	high
Plasma Protein Binding	%PPB	logKa = log10((100% - PPB)/PPB), with PPB in %. See also [8]	low
Solubility	Solubility at given pH, separate task for each pH reported	Log10 for the molar solubility. Solubility values in g/l need to be converted using the solute's molecular weight.	low
logP	logP	Unitless, no conversion necessary	high
logD	logD at given pH, separate task per each pH reported	Unitless, no conversion necessary	high
CYP time dependent inhibition	K_{obs}	Log10(K_{obs} [min^{-1}])	high
Microsomal and hepatocyte clearance	clearance	Log10 (CL_{int}) with CL_{int} in µL/min*mg. It is understood that not all partners can easily convert their read-outs to this desired target representation. In this case it is till recommended to log-transform the available clearance read-out	high
Standardized in vivo-PK	Clearance	Log10(Clearance measured in [mL/min/kg]). Use only data from intravenous (i.v.) dosing	high
	Bioavailability	Log10(Bioavailability) from oral dosing	low
	Vss (volume of distribution)	L/kg, log10 scaled	high
	MRT (mean residence time)	h, log10 scaled	low

Table 5: Units and scaling for experimental ADME readouts.

3.3.4 Scaling of single concentration HTS assays

It is expected that the pharma partners will access instrument read-outs that are normalized to the neutral (low) and, if available active (high) controls on the same plate. These readouts will typically have the form of %activity, and ideally have undergone further QC procedures such as Gubler *et al.*[9]

The activity values obtained in this way are then re-normalized across the whole sample domain of the assay, that is all measured samples, using the robust Z-score also referred to as rscores. For each data representation possibility, we use the r-scored values of HTS data using the formula below:

$$\text{rscore}_i = (x_i - \text{median}(X)) / \text{mad}(X)$$

xi: the raw activity value of sample i

X: the percentage activity value vector of the assay. This will typically be percentage activity in case there is a neutral and an active control.

mad: median absolute deviation

The presence of read-outs can be ignored for the purpose of calculating median and mad.

Typically, these assays have also one direction of interest, in which the desired activity can be found. Single partner studies demonstrated that classification using absolute rscore values worked well, providing that setting the direction of interest is possible though not strictly required.

A script to convert the input HTS data into r-scores will be provided. This script will also annotate samples as frequent hitters based on the rscore readout and the method by Nissink *et al.*[10] Partners can use their own frequent hitter annotation instead.

3.4 Definition of Expert Thresholds

For the classification models, most classification tasks will be generated using automatically defined thresholds during the MELLODDY_Tuner processing, as described below in §4.2.4 by analysing the task value distribution. However, partners have the possibility to pre-define thresholds based on expert knowledge. These thresholds will be added to the T0 file. The number of expert defined thresholds must not exceed five for each assay. Reasons for defining expert thresholds can be:

- Assay read-out thresholds triggering regulatory consequences, such as in safety panel assays
- Replication of assay reads-out thresholds used in internal models
- Harmonization of thresholds among partners for assays contracted out to CROs and used identically by multiple partners

Expert thresholds must match in unit and scale the transformed activity data.

In order to ensure a uniform representation of assays undergoing catalogues assay fusion, for all assays of the type CATALOG-PANEL the following expert thresholds will be enforced by MELLODDY-Tuner and override any expert thresholds set in the input: 5 (10 µM), 6 (1 µM), and 7 (0.1 µM). Further auxiliary thresholds like 4.5 (31 µM) or others will be included in the given reference file (Tcat) if benefical.

3.5 Binary Task Values

It is possible to present assay results in binary form. This needs to be indicated by a flag in the T0 file. For these assays, the thresholding step in MELLODDY-Tuner is bypassed and the data is directly presented as classification tasks. Because of the discrete nature, no regression is possible for such tasks.

There are two use cases for this:

- Auxiliary pseudolabel data, which is generated in binary form.
- Results from assays which are only reported in discrete categories.

For all other assays, which report their data as continuous readout, this mechanism should **not** be used. If desired, thresholding of continuous valued assay data can be controlled by using the expert threshold mechanism described above (§ 3.4).

3.6 Replicate Aggregation on Sample Level

Background

Different types of replicates can occur during structure activity data processing:

- ***Technical replicates:*** *In some assays, all measurements are systematically done in replicates for each submitted sample. Such technical replicates are typically already aggregated at the stage the assay results are reported into the data warehouse.*
- ***Sample replicates:*** *Sample replicates result from repeated independent submission of the same compound sample to the assay.*
- ***Structure replicates****: Measurements of different samples with the same attributed chemical structure. Detecting this in a consistent manner across partners requires uniform structure standardization to be used at each partner's side*
- ***Descriptor replicates****: Result from measurements of samples with different chemical structures but having the same descriptor vectors. This type of replicates is undesirable, as it results from the inability of the descriptor to encode all relevant structural information required for their distinction. In the case of descriptors not encoding stereochemistry such as the ECFP variants used, this means that results on different stereoisomers will be descriptor replicates. Even if such replicates were not explicitly aggregated during data preparation, the ML algorithm would implicitly aggregate the data on the items indistinguishable to it.*

Pharma partners are expected to aggregate replicates at the sample level. Later in the work-flow aggregation will take place at the descriptor level. As replicate read-outs taken on the same sample are expected to give the same result, this is the stage where best to deal with diverging measurements. The following principle apply:

- Aggregations of technical replicates made before data warehouse entry are treated as a single result and not undone.
- Measurements resulting from independent, repeated submissions of the same sample to an assay are aggregated according to internal rules of the pharma partners, which will typically be averaging.
    - In case of divergent measurements partners should remove all observations if there is no knowledge available, which replicate is wrong (respectively right). If partners are in possession of knowledge, which observation is the correct one, they may remove the erroneous replicate, as partners are in general encouraged to remove observations they know to be erroneous or be problematic.

3.7 Preparation for Time-Gated Analysis

This section describes the necessary steps for time gated analysis. There are two steps required:

- **Addition of time stamps:** Two time stamps can be considered:
    - **Structure registration dates**: necessary to define the temporal data set based on a pure compound wise split. The registration dates of a sample should represent the minimum registration date at which any sample of the same structure was registered at (i.e. the oldest date at which the structure has appeared in data warehouses). Sample A structure registration dates can be assigned to each sample included in T2.
    - **Experimental timestamps:** necessary to define temporal data sets on the basis of each modelled assay. Aggregated experimental timestamps should represent the oldest date at which any experiment with the sample structure was performed (i.e. the first experiment in

this assay with any sample of the same structure). Experimental timestamps can be included in T1.

- **Ability to map year 2 tasks to year 3 tasks**: The most straightforward way to achieve this, to use for each internal source assay the same input assay ID in year 2 and year 3. Otherwise a mapping table between year 2 and year3 can be used.

3.8 File Formats

3.8.1 Assay Metadata File (T0)

input_ assay_id	assay_type	use_in_ regression	Is_binary	expert_ threshold_ 1	... _2	... _3	... _4	... _5	direction	catalog_ assay_ id	parent_ assay_id
3855277	OTHER	True	False								
3855278	OTHER	True	False								
3855279	CATALOG-PANEL	True	False	5.0						12	
3855298	ADME	True	False	2.0					high		
9999998	AUX_PL	False	True								3855277
9999999	AUX_HTS	False	False								

Table 6: Example of a T0 table. Even if the expert thresholds for input_assay_ID 3855279 had not been present or different than those shown here, MELLODDY Tuner would have imposed these thresholds for CATALOG-PANEL assays

This comma separated file contains the initial assay metadata required for pre-processing and machine learning.

- **input_assay_id** represents the identifier for the assay (integer format), and needs to be unique in this file. If the year 3 MELLODDY Tuner input should be suitable for time gated analysis (optional!), then either the same input_assay_id needs to be used for each source assay in year 2 and year 3 input, or at least a mapping table needs to exist.

- **assay_type** must be one of the following “NON-CATALOG-PANEL”, “CATALOG-PANEL”, “ADME”, “OTHER”, “AUX_HTS”, or “AUX_PL”.

- **use_in_regression**: Can be either “True” or “False” and indicates whether a task should be included in the regression dataset. In the year 3 run, the partners are asked to include only such tasks into regression, where there is a chance that a meaningful regression model can be obtained. If this is left empty, True is assumed, unless the task is marked as binary. For tasks of type AUX_HTS and AUX_PL, or tasks marked as binary use_in_regression = “False” is enforced. Please note that it is permitted set use_in_regression = True for CATALOG-PANEL tasks, despite the fact, that catalogue fusion is only used for classification. In the regression arm of the data preparation, this task will be converted to a “NON-CATALOG-PANEL” task.

- **Is_binary**: Can be either Tue or False, with False being the default. True indicates that the data for this assay is presented in binary form already. Is_binary = True implies automatically use_in_regression = false. AUX PL type tasks are presented already in binary form.

- **expert_threshold_1, expert_threshold_2, expert_threshold_3, expert_threshold_4, expert_threshold_5** : Columns for providing optional expert defined thresholds (see 3.4). Either empty, or a single floating point number, indicating a threshold. Expert Thresholds for assays of type “CATALOG-PANEL” are ignored.

- **direction**: Encodes the direction at which end of the scale compounds of interest to be predicted are located (see also §3.3.3 and §3.3.4). This must be either “high” or “low”. Filling this column is mandatory for the assay type ADME, optional for AUX_HTS. For the main activity endpoints in the categories “NON-CATALOG-PANEL”, “CATALOG-PANEL” and “OTHER” this is not interpreted, as here always the high end of the pIC50-type scale is considered of interest.

- **catalog_assay_id**: Column referencing a catalogue assay ID. Must match an ID from a reference file **Tcat** (seel below in Table 14) made accessible to pharma partners. This column is mandatory to be filled for assays of type “CATALOG-PANEL”. Each catalog_assay_id may be mapped only on exactly one input_assay_id

- **parent_assay_id**: For auxiliary assays then can be linked to a parent assay it is possible to add here the input assay_id for that task

The presence of other columns is tolerated, they are carried through, and only removed at the last stage when the data files for ML are written. All assays are expect to be listed in the T0 file, including auxiliary tasks and assays. MELLODDY-Tuner accepts multiple T0 files as input, which will then be concatenated. Across the T0 files, the input_assay_id must be unique.

3.8.2 Activity Data File (T1)

input_compound_id	input_assay_id	standard_qualifier	standard_value
3950540	3855277	=	9.58
3955505	3855277	=	9.38
3896638	3855277	=	9.24
3946901	3855277	=	9.10
3927895	3855277	=	8.98
3984387	3855277	=	8.56
3905582	3855277	=	8.50
3935776	3855277	=	8.49
3983937	3855277	=	8.26
3901587	3855277	=	8.21
85606	3855277	=	8.19
3967683	3855277	=	8.11
3935776	3855277	=	8.03
3913062	3855277	=	7.75
3958494	3855277	=	7.60
3958494	3855277	=	7.55
3901211	3855277	=	7.38
3927895	3855277	=	7.14
3913062	3855277	=	6.10
3984387	3855278	=	6.93
3905582	3855278	=	6.89
3955505	3855278	=	6.65
3935776	3855278	=	6.51
3935776	3855278	=	6.50
3896638	3855278	=	6.48
3927895	3855278	=	6.48
3950540	3855278	=	6.46
85606	3855278	<	6.31
3958494	3855278	=	6.29
3946901	3855278	=	6.28
3967683	3855278	=	6.20
3901211	3855278	=	6.15
3983937	3855278	=	6.10
3913062	3855278	=	6.03
3901587	3855278	=	6.01
3958494	3855278	=	6.00
3927895	3855278	=	5.36
3913062	3855278	=	4.77
3950540	3855279	>	7.00
3896638	3855279	>	7.00
3984387	3855279	=	6.95
3946901	3855279	=	6.70
3946901	3855279	=	6.68
3935776	3855279	=	6.60
3901587	3855279	=	6.35
3955505	3855279	=	6.28
85606	3855279	=	6.17
3927895	3855279	=	6.16
3983937	3855279	=	5.54
3901211	3855279	=	5.44

input_compound_id	input_assay_id	standard_qualifier	standard_value
3958494	3855279	=	5.40
3913062	3855279	=	5.33
3905582	3855279	<	5.00
3901587	3855298	=	1.30
3955505	3855298	=	2.03
3941818	3855298	=	2.55
3901587	9999998	=	1
85606	9999998	=	-1
3967683	9999998	=	1
3935776	9999998	=	1
3913062	9999998	=	-1
3958494	9999998	=	-1
3905582	9999998	=	1
3901211	9999998	=	-1
3927895	9999998	=	-1
3913062	9999998	=	-1
3935776	9999999	=	-4.18
3958494	9999999	=	-1.60
3927895	9999999	=	-1.29
3913062	9999999	=	-0.95
85606	9999999	=	-0.88
3901587	9999999	=	-0.78
3905582	9999999	=	-0.21
3896638	9999999	=	0.10
3901211	9999999	=	0.65
3946901	9999999	=	0.76
3955505	9999999	=	1.05
3984387	9999999	=	1.68
3983937	9999999	=	2.26
3950540	9999999	=	2.67
3967683	9999999	=	2.89

Table 7: Example Data: Input activity data (T1 as in Figure 1: General data preparation workflow. The labels T0 to T11 in this figure refer to example data tables.), with unit scale and transformations applied

The following columns need to be present:

- **input_compound_id**: refers to input_compound_id in T2
- **Input_assay_id**: refers to input_assay_id in T0
- **standard_qualifier**: The modifiers for censored data are to be written into this column and are limited to ['<', '<=', '<<', '>', '>=', '>>', '=', '~']. Empty values in the activity modifier column are interpreted as equals. In MELLODDY-Tuner the qualifiers will be mapped to <, =, or >.
- **standard_value**: the numeric value of the observation, scaled according to §3.2.3. For task that are declared as binary tasks using the is_binary flags in the T0 file, the values need to be either 1 or -1, all other values are treated as an error. 1 should correspond to the (typically minority) class for which the area under the precision recall curve is determined.

The T1 format is used for all assays and tasks, including auxiliary data. If desired MELLODDY-Tuner can read in multiple T1 files which will then be concatenated.

3.8.3 Structure File (T2)

This table contains the chemical structures for all compounds covered by the activity data file, including compounds having only auxiliary data. The following columns are expected to be present:

- **Input_compound_id**: a unique identifier as integer for each compound sample
- **smiles**: smiles representation of the chemical structure associated with the sample as present in the company data warehouse without additional standardization

input_compound_id	smiles
3896638	Cn1c(SCCCN2CC[C@]3(C[C@@H]3c4ccc(cc4)C(F)(F)F)C2)nnc1c5ccsc5
3901211	Cc1ncoc1c2nnc(SCCCN3CC[C@@]4(C[C@@H]4c5ccccc5)C3)n2C
3905582	Cn1c(SCCCN2CC[C@]3(C[C@@H]3c4ccc(cc4)C(F)(F)F)C2)nnc1c5csnn5
3913062	Cc1ncoc1c2nnc(SCCCN3CC[C@]4(C[C@@H]4c5ccccc5)C3)n2C
3941818	Cn1c(SCCCN2CC[C@]3(C[C@@H]3c4ccc(cc4)C(F)(F)F)C2)nnc1c5ccc(cc5)c6occn6
3946901	Cn1c(SCCCN2CC[C@]3(C[C@@H]3c4ccc(cc4)C(F)(F)F)C2)nnc1c5ccccn5
3950540	Cn1c(SCCCN2CC[C@]3(C[C@@H]3c4ccc(cc4)C(F)(F)F)C2)nnc1c5ccc(cc5)C#N
3958494	Cc1ncoc1c2nnc(SCCCN3CC[C@]4(C[C@H]4c5ccccc5)C3)n2C
3967683	Cn1c(SCCCN2CC[C@]3(C[C@H]3c4ccc(cc4)C(F)(F)F)C2)nnc1c5ccccn5
3983937	Cc1ncoc1c2nnc(SCCCN3CC[C@@]4(C[C@H]4c5ccccc5)C3)n2C
3984387	Cn1c(SCCCN2CC[C@]3(C[C@H]3c4ccc(cc4)C(F)(F)F)C2)nnc1c5csnn5
3901587	Cc1ncoc1c2nnc(SCCCN3CC[C@@]4(C[C@@H]4c5ccc(cc5F)C(F)(F)F)C3)n2C
3927895	Cc1ncoc1c2nnc(SCCCN3CC[C@]4(C[C@@H]4c5ccc(cc5F)C(F)(F)F)C3)n2C
3935776	Cc1ncoc1c2nnc(SCCCN3CC[C@]4(C[C@H]4c5ccc(cc5F)C(F)(F)F)C3)n2C
3955505	Cc1ncoc1c2nnc(SCCCN3CC[C@@]4(C[C@H]4c5ccc(cc5F)C(F)(F)F)C3)n2C
85606	O=C(N[C@@H]1CC[C@@H](CCN2CCc3cc(ccc3C2)C#N)CC1)c4ccnc5ccccc45

Table 8: Example Data: Input smiles data (T2 as in Figure 1: General data preparation workflow. The labels T0 to T11 in this figure refer to example data tables.)

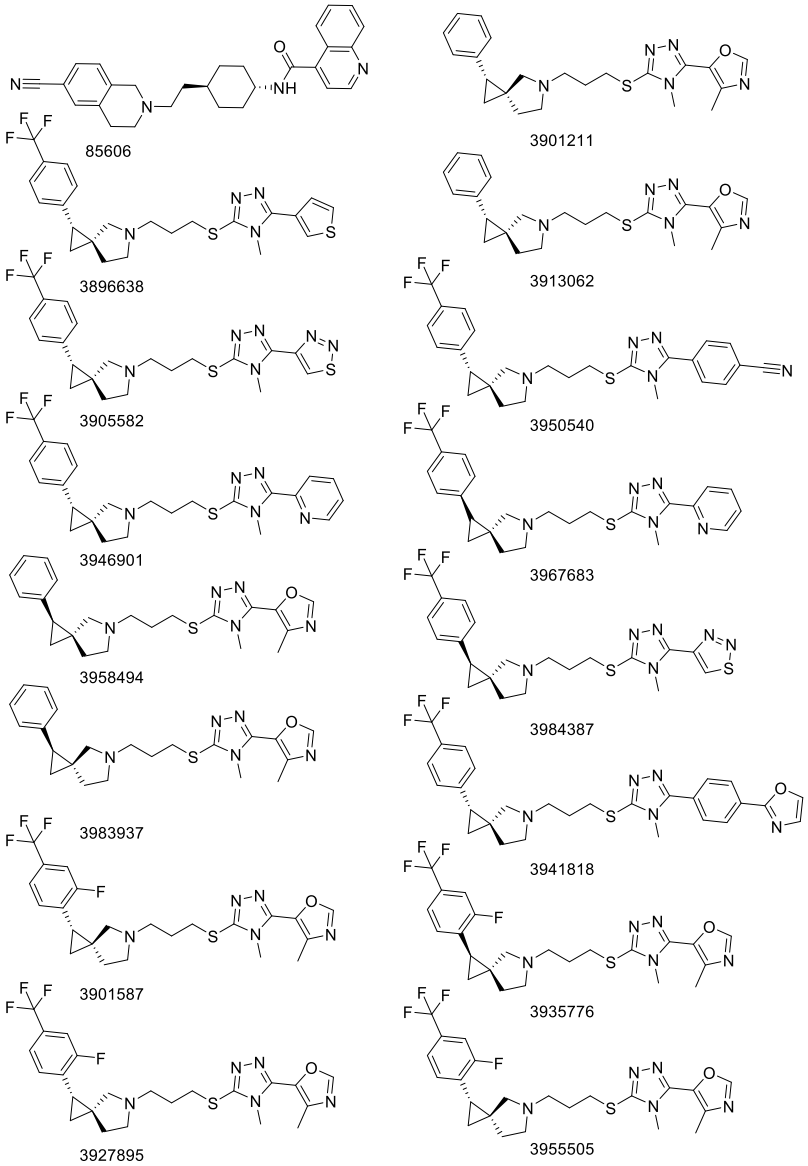

Figure 3: Structure depictions for example structures from Table 8 (T2 example)

The T2 structure file may contain more compounds than actually referenced in the T1 file(s). Non-reference structures will be dropped. This enables to use the same comprehensive structure file as input for the MELLODDY-Tuner run with and without auxiliary data.

3.9 Generation of auxiliary pseudo-labels from high content data

The overall process has been described in 2.1.2. In order to execute this, code to perform the necessary steps has been provided by Iktos as a code package separate from MELLODDY-Tuner. This code expects the following as input:

1. A feature matrix file from the high content experiment. The feature matrix must be of the form of a csv file, with the left most index column containing the input_compound_id used in the structure file followed by a column for each feature. The header of the column is ignored. See also the example in Figure 4.

2. A structure file in T2 format (see 3.8.3) containing the structure for each entry in the feature matrix

3. The readily computed MELLODDY-Tuner output from the main-task data only.

While some overlap is needed between the input_compound_IDs of the feature matrix and the input_compound_IDs in the input used to generated the main task MELLODDY-Tuner output, additional compounds may be contained in the feature matrix. The output of the pseudolabel generation code is an incremental T0, T1 and T2 file for the pseudolabel data.

	Nuclei_Texture_InfoMeas2_Hoechst_3_0	Cells_Neighbors_NumberOfNeighbors_Adjacent	Cytoplasm_RadialDistribution_RadialCV_Hoechst_4of4	Cyt
0	0.095647	0.00000	0.171480	
1	0.185306	-0.67449	0.380434	
2	0.057712	-0.67449	0.138456	
3	-0.009696	-0.67449	-0.259146	
4	0.070429	0.00000	0.214619	
...	...	...	...	
995	-0.099112	0.00000	-0.068375	
996	-0.008187	0.00000	0.335033	
997	-0.032211	-0.67449	0.040137	
998	-0.101553	0.00000	-0.129782	
999	0.013239	0.00000	0.056723	

Figure 4: Example of feature matrix file

Due to the dependency of the pseudolabel generation on the main-task MELLODDY-Tuner output the overall process needs to be as follows:

1. Prepare the main-task data in order to obtain the main task T0 and T1 files. Prepare a comprehensive T2 file including all structure from main tasks, auxiliary HTS tasks, and auxiliary feature matrices. Process the main task data with MELLODDY-Tuner

2. Run the pseudo-label generation code. If there are multiple feature matrices to be included, this needs to be done separately for each feature matrix to be included

3. Run MELLODDY-Tuner on the T0, T1, and T2 input with auxiliary data, sing the option of MELLODDY Tuner to accept multiple files for T0 and T1

4 Processing through MELLODDY-Tuner

4.1 Structure processing

4.1.1 Standardization of Input Structures

In order to ensure a consistent representation of the chemical structures, the input structures read in from T2 are standardized.

The following operations are included:

- Stripping counterions of salts
- Removing stereo information
- Neutralizing the molecule
- Tautomer canonization

This process is done using the MolStandardize module from RDkit (https://www.rdkit.org/) in its default settings unless specified otherwise:

```
from  rdkit.Chem.MolStandardize.rdMolStandardize import
CleanupParameters, TautomerEnumerator


my_standardizer = MolStandardize.standardize.Standardizer()
mol = my_standardizer.charge_parent(mol)  # standardize molecules using
mol = my_standardizer.isotope_parent(mol)
mol = my_standardizer.stereo_parent(mol)
mol = my_standardizer.tautomer_parent(mol)
mol_clean = my_standardizer.standardize(mol)
```

Depending on the parameter settings, generation of the stereo parent can be switched off in the code to retain stereoinformation, which may be desirable in subsequent years. As in the year one run, however stereoinformation will be stripped, as achiral fingerprints will be used. The tautomer standardizer follows largely the rules described by Sitzmann *et al.*[11] From the 2020_09 version onwards RDkit has the option to preserve steoreochemistry on sp3 carbons undergoing tautomerism. This option is used.

Structures having > 100 non-H atoms are treated as failed, saved in a separated file and will be removed from the data set. A table of failing structures is written out for inspection.

In order to avoid excessive time spent on the standardization of individual structures, the enumeration of tautomers is limited to 1000 and to treat molecules exceeding this enumeration limit as failed and, therefore, to exclude them.

The resulting standardized structures are saved in two file formats (csv and numpy file (.npy)) in order to allow re-using already standardized structures, as this is the most time-consuming step.

4.1.2 Scaffold based Fold Assignment

Compounds are assigned to folds using their scaffold. The general process is as follows

1. Prune all terminal sidechains to obtain the Murcko scaffold[12]
2. Decompose the Murcko scaffold into subscaffolds by breaking linker bonds, but without decomposing fused or bridged ring systems[13]
3. From the so obtained scaffold, extract the most representative scaffold using the following precedence rules:
    a. The number of rings is closest to 3 (minimize abs(numrings – 3))
    b. Scaffold precedence rules according to Scaffold Tree[14]
    c. Alphabetic sorting order of the scaffold's canonical smiles
4. The canonical smiles of the scaffold identified in 3. is then hashed together with the secret key. The digest of this hashing procedure seeds a random number generator used to generate a random integer to indicate the fold. As the desired number of folds is 5, integers in range from 0-4 will be generated.

For acyclic molecules the scaffold is empty, and an empty string is passed as the scaffold smiles to the hashing algorithm. Consequently, all acyclic structures will end up in the same fold.

4.1.3 Calculation of ECFP Fingerprints

For each chemical structure an Extended Connectivity Fingerprint (ECFP)[15,16] is calculated as input for the ML algorithm. The original structures will not be presented to the ML algorithm. To calculate these fingerprints the RDKit implementation of ECFPs called “Morgan Fingerprint” will be used.

The following parameters must be defined when using Extended Connectivity Fingerprints

- **Type (Bit or Count)**: Whether to use only the binary information of fragment presence or absence, or the fragment count
- **Size (Radius)**: The maximal radius around the central atom of the encoded fragments
- **Modification** (Length reduction options):
  - **Folding**: Reduction of the fingerprint length by hashing each fingerprint bit to an integer in the range of the desired length
  - **Low Frequency Truncation**: Retain only the L most frequent fingerprint bits with L being the desired length.
  - **Tail compression**: Compression of the low frequently occurring fingerprint bits (e.g. taking 2000 neurons for the 2000 most frequent features, and much less neurons for the remaining features).
- **Length**: length of the resulting output
- **Chirality**: Whether to include chirality or not

Both Low Frequency Truncation and tail compression require the estimation of bit frequencies across the compound space of all partners. These steps therefore need to be incorporated into the federated ML code and cannot be done by each partner individually at the data preparation stage. In contrast to this, fingerprint folding will be executed at each partner side within the data processing stage.

The final fingerprint setup is described in Table 9

Parameter	Setup A
Type	Bit
Size (Radius)	ECFP6 (3)
Modification	Folded
Length	32000
Chirality	No
Bit permutation	Yes

Table 9: ECFP Setup

Structures which fail in the generation of the fingerprint or produce an empty fingerprint will be considered as failed and will be removed from the dataset. The processing script will write a file of those structures failing processing either at the standardization or the descriptor generation stage.

The fingerprints will be subsequently scrambled. In order to do this, a bit permutation map is generated which contains a permutation of the possible bit indices in the range from 0 to the specified fingerprint length. This map is then used to map the original bit indices into the permuted bit index space.

4.1.4 Assignment of a Unique Descriptor ID

For those compound structures, which passed the steps so far, a unique identifier is generated for each unique ECFP descriptor / fold_id combination. The decriptor_vector / fold_id combination is used, as in very rare cases molecules with two different scaffolds can result in the same descriptor and any ambiguity about the fold allocation can in such cases be avoided. This descriptor_id will be used for replicate aggregation on descriptor level. A mapping table is written, that maps input_structure_id to descriptor_ids and fold_ids.

input_compound_id	fold_id	descriptor_vector_id
3901587	0	1
3927895	0	1
3935776	0	1
3955505	0	1
3950540	0	2
3905582	0	3
3984387	0	3
3941818	0	4
3946901	0	5
3967683	0	5
3896638	0	6
3901211	0	7
3913062	0	7
3958494	0	7
3983937	0	7
85606	2	8

Table 10: Example Data: Mapping between input compound identifiers and descriptor_vector_ids (corresponds to T5 in Figure 1: General data preparation workflow. The labels T0 to T11 in this figure refer to example data tables.).

Likewise, a table containing the unique descriptor_id alongside with the permuted ECFP and the fold ID is written. This table contains one row per unique **descriptor_id**. The fingerprint is provided as list of bits encoded as json string.

descriptor_vector_id	fold_id	fp_on_bits
1	0	[754, 841, 926, 1039, 1515, 1673, 1787, 1804, 2000, 2226, 2947, 3534, 3881, 4444, 4708, 5247, 5548, 5969, 6009, 6695, 7326, 8094, 8272, 8428, 8438, 8648, 8737, 8833, 9231, 9383, 9639, 9768, 10507, 10819, 11361, 11605, 11607, 12122, 12260, 12418, 12559, 13043, 14721, 14723, 14847, 15498, 15560, 15663, 15880, 15984, 16164, 16278, 16756, 17044, 17083, 17293, 17552, 17850, 18297, 18304, 19569, 20206, 20329, 20410, 20528, 21099, 22068, 22156, 22789, 23169, 23394, 23469, 24273, 24326, 24351, 26070, 26490, 26624, 26919, 27003, 27066, 27207, 27645, 28908, 29119, 29303, 29539, 29609, 29810, 29955, 30140, 30456, 31097, 31112, 31795]
2	0	[841, 870, 906, 926, 1515, 1673, 1787, 1791, 1804, 2000, 2640, 3881, 3894, 4708, 5247, 5387, 5778, 5969, 6009, 7248, 7326, 8094, 8272, 8610, 8648, 8737, 8871, 9064, 9353, 9383, 9450, 9639, 10182, 11605, 11607, 12122, 12260, 12418, 12962, 13043, 14561, 14567, 14721, 14723, 14847, 14897, 14958, 15232, 15498, 15663, 15880, 16164, 16756, 17044, 17293, 18297, 20410, 22068, 22156, 22399, 22641, 22789, 23169, 23394, 23469, 23920, 24200, 24351, 26070, 26490, 26582, 26624, 27003, 27158, 27170, 27207, 27645, 28251, 28908, 29023, 29303, 29539, 29605, 29609, 29810, 30456, 31097, 31112, 31453]
3	0	*<on bit list as json string>*
4	0	*<on bit list as json string>*
5	0	*<on bit list as json string>*
6	0	*<on bit list as json string>*
7	0	*<on bit list as json string>*
8	2	*<on bit list as json string>*

Table 11: Example Data: Unique descriptor vector ids (corresponds to T6 in Figure 1: General data preparation workflow. The labels T0 to T11 in this figure refer to example data tables.). The fold_id is a mock_up only, and the descriptor bits shown here are not permuted

4.2 Processing of Activity Data

4.2.1 Validation of the Assay Metadata (T0) File

The Assay Metadata file T0 is read and validated to confirm that the input complies with the file specifications described in §3.8.1. A Boolean column “is_auxiliary” is created that contains the information whether an assay (or later on derived task) is auxiliary. This column is initialized with False for all assay_types except “AUX_HTS” and “AUX_PL”.

4.2.2 Remove values out of credible value range

For each assay type credibility ranges can be defined in the parameter file. Observations outside of the credibility ranges are removed from the T1 file. For assays presented as binary tasks (flag is_binary in T0 equal True) this credibility range check is executed in the way, that only values of 1 and -1 are accepted.

4.2.3 Replicate Aggregation on Descriptor Level

The T5 table mapping input_compound_id to descriptor ID is joined to the table with the activity values (T1). This allows now the identification of replicate measurements on the descriptor_id level and their aggregation.

The aggregation rules depend on the assay type:

assay_type	Aggregation Rule
NON-CATALOG-PANEL CATALOG-PANEL OTHER	Retain the value corresponding to the highest activity • If there are observations without an < qualifier, use the observation with the highest numerical value from those observations • Otherwise simply use the observation with the highest numerical value
ADME	Retain the median result
AUX_HTS	• If the direction is “high” retain the largest value • If the direction is “low” retain the smallest value • If no direction is given retain the observation with the maximal absolute value
AUX_PL	These tasks will be exclusive presented as binary tasks, so that aggregation rules for binary tasks apply.

Table 12: Aggregation rules

Irrespective of the assay_type, for assays marked as binary (is_binary = True in the T0 file) the following rules apply:

- The most common class label is retained.
- In case of ties, where -1 and +1 are equally common, +1 is retained, as +1 is assumed to be the minority class.

An example of the resulting T4r table can be found below (Table 13)

input_assay_id	descriptor_id	fold_id	*input_compound_id*			standard_qualifier	standard_value
			3927895	=	*7.14*		
			3935776	=	*8.03*		
			3901587	=	*8.21*		
			3935776	=	*8.49*		
			3927895	=	*8.98*		
3855277	1	0	*3955505*	=	*9.38*	=	9.38
3855277	2	0	*3950540*	=	*9.58*	=	9.58
			3905582	=	*8.50*		
3855277	3	0	*3984387*	=	*8.56*	=	8.56
			3967683	=	*8.11*		
3855277	5	0	*3946901*	=	*9.10*	=	9.10
3855277	6	0	*3896638*	=	*9.24*	=	9.24
			3913062	=	*6.10*		
			3901211	=	*7.38*		
			3958494	=	*7.55*		
			3958494	=	*7.60*		
			3913062	=	*7.75*		
3855277	7	0	*3983937*	=	*8.26*	=	8.26
3855277	8	2	*85606*	=	*8.19*	=	8.19
			3927895	=	*5.36*		
			3901587	=	*6.01*		
			3927895	=	*6.48*		
			3935776	=	*6.50*		
			3935776	=	*6.51*		
3855278	1	0	*3955505*	=	*6.65*	=	6.65
3855278	2	0	*3950540*	=	*6.46*	=	6.46
			3905582	=	*6.89*		
3855278	3	0	*3984387*	=	*6.93*	=	6.93
			3967683	=	*6.20*		
3855278	5	0	*3946901*	=	*6.28*	=	6.28
3855278	6	0	*3896638*	=	*6.48*	=	6.48
			3913062	=	*4.77*		
			3958494	=	*6.00*		
			3913062	=	*6.03*		
			3983937	=	*6.10*		
			3901211	=	*6.15*		
3855278	7	0	*3958494*	=	*6.29*	=	6.29
3855278	8	2	*85606*	<	*6.31*	<	6.31
			3927895	=	*6.16*		
			3955505	=	*6.28*		
			3901587	=	*6.35*		
3855279	1	0	*3935776*	=	*6.60*	=	6.60
3855279	2	0	*3950540*	>	*7.00*	>	7.00
			3905582	<	*5.00*		
3855279	3	0	*3984387*	=	*6.95*	=	6.95
			3946901	=	*6.68*		
3855279	5	0	*3946901*	=	*6.70*	=	6.70
3855279	6	0	*3896638*	>	*7.00*	>	7.00
			3913062	=	*5.33*		
			3958494	=	*5.40*		
			3901211	=	*5.44*	=	5.54

3855279	7	0	*3983937*	=	*5.54*		
3855279	8	2	*85606*	=	*6.17*	=	6.17
			3901587	=	*1.30*		
3855298	1	0	*3955505*	=	*2.03*	=	1.67
3855298	4	0	*3941818*	=	*2.55*	=	2.55
			3901587	=	*1*		
			3935776	=	*1*		
9999998	1	0	*3927895*	=	*-1*	=	1
9999998	3	0	*3905582*	=	*-1*	=	-1
9999998	5	0	*3967683*	=	*1*	=	1
			3913062	=	*-1*		
			3958494	=	*-1*		
			3913062	=	*-1*		
9999998	7	0	*3901211*	=	*-1*	=	-1
9999998	8	2	*85606*	=	*-1*	=	-1
			3935776	=	*-4.18*		
			3927895	=	*-1.29*		
			3901587	=	*-0.78*		
9999999	1	0	*3955505*	=	*1.05*	=	-4.18
9999999	2	0	*3950540*	=	*2.67*	=	2.67
			3905582	=	*-0.21*		
9999999	3	0	*3984387*	=	*1.68*	=	1.68
			3946901	=	*0.76*		
9999999	5	0	*3967683*	=	*2.89*	=	2.89
9999999	6	0	*3896638*	=	*0.10*	=	0.10
			3958494	=	*-1.60*		
			3913062	=	*-0.95*		
			3901211	=	*0.65*		
9999999	7	0	*3983937*	=	*2.26*	=	2.26
9999999	8	2	*85606*	=	*-0.88*	=	-0.88

Table 13: Example for a T4r table with aggregated activity readouts. Columns in italics are intermediate results only and not expected to be present, but shown here for illustration only.

After replicates have been aggregated, the activity data from concentration response assay (type OTHER and PANEL) is rounded to two decimal places. Given that the data is log-scaled (see 3.3.1), this does not lead to a loss of accuracy, given the experimental uncertainty of the underlying values. Since ADME assay read-outs are less uniform and can be present in either log- or linear-scale they will not be rounded.

4.2.4 Generation of Classification tasks

As a first step, the main thresholds and possible auxiliary thresholds need to be defined. How this is done depends on the assay type:

- **OTHER**, **NON-CATALOG-PANEL**: If no expert threshold has been defined, the main threshold will be determined as follows:

    1. Loop over range of log-unit activity values: {8.0, 7.0 ,6.0, 5.0, 4.7 ,4.4}
    2. Select the largest threshold where the fraction of actives is >= 20 % and the quorum of 25 actives and 25 inactives is met.
    3. If no threshold can be identified that way, the median is used as fall-back option

Two auxiliary "sandwich" thresholds will be added to the edges of the value range (0.5 log-unit above the largest threshold and 0.5 log-unit below the smallest thresholds) if not more than three main thresholds are defined. If there are more than three main thresholds defined, no auxiliary thresholds are created.

- **CATALOG-PANEL:** For this assay type a reference table is consulted that contains for each permitted catalog_assay_id a list of the thresholds to be used alongside with the associated catalog_task_ids.

- **ADME:** If there are no expert threshold defined, then the median and a quantile threshold will be used as main threshold. For assays with direction "high" the 75th percentile will be added, for assay with the direction "low" the 25th percentile will be added

- **AUX_HTS**: As these are auxiliary assays only, only one auxiliary threshold is defined at a fixed R-score cut-off which is defined in the parameter file

catalog_assay_id	Is_auxiliary	threshold	threshold_method	catalog_task_id
12	False	5.0	expert	101
12	False	6.0	expert	102
12	False	7.0	expert	103
12	True	4.5	aux_low	104
12	True	7.5	aux_high	105

Table 14: Example for a Tcat table listing the unique catalogue task IDs for all permitted comibnations of catalog assay_id and threshold. All catalog assays will use the same thresholds.

For determining threshold locations, the censored data are converted with an offset in order to calculate quantiles. The result of this is a **T3c** table specifying the classification thresholds. Classification tasks inherit the relevant metadata attributes of their assays as specified in T0.

For tasks specified in T0 as binary (is_binary = True) this step is skipped, and each of those tasks is directly treated as classification task.

Classification_ task_id	input_ assay_id	assay_type	use_ln_ regression	is_binary	ls_ auxiliary	threshold	threshold_ method	direction	catalog_ assay_id	catalog task_i
1	3855277	OTHER	True	False	False	9.0	fixed_ adapative			
2	3855277	OTHER	True	False	True	8.5	aux_low			
3	3855277	OTHER	True	False	True	9.5	aux_high			
4	3855278	OTHER	True	False	False	6.0	fixed_ adaptive			
5	3855278	OTHER	True	False	True	6.5	aux_high			
6	3855279	CATALOG-PANEL	True	False	False	5.0	expert		12	101
7	3855279	CATALOG-PANEL	True	False	False	6.0	expert		12	102
8	3855279	CATALOG-PANEL	True	False	False	7.0	expert		12	103
9	3855279	CATALOG-PANEL	True	False	True	4.5	aux_low		12	104
10	3855279	CATALOG-PANEL	True	False	True	7.5	aux_high		12	105
11	3855298	ADME	True	False	False	2.0	expert	high		
12	9999998	AUX_PL	False	True	True					
13	9999999	AUX_HTS	False	False	True	3.0	fixed			

Table 15: Table with classification thresholds defining the classification tasks (T3c)

In a second step these thresholds are applied to the activity data from the **T4r** table. The following rules apply:

- If a threshold is for an assay of type OTHER, NON-CATALOG-PANE, or CATALOG-PANEL, or the assay direction is “high”:
    - all observations having a numerical value ≥ threshold and do not have a qualifier < get class label +1 (“active”)
    - all observations having a numerical value < threshold get class label -1 (“inactive”)
    - all observations having a numerical value ≥ threshold and have a qualifier < cannot be decided unambiguously and will be removed
- If the assay direction is “low”:
    - all observations having a numerical value ≤ threshold and do not have a qualifier > get class label +1 (“active”)
    - all observations having a numerical value > threshold get class label -1 (“inactive”)
    - all observations having a numerical value ≤ threshold and have a qualifier > cannot be decided unambiguously and will be removed
- If a threshold is for an assay of type AUX_HTS and no direction is given:
    - All observations with an absolute numerical value ≥ threshold get the class label +1 (“active”)
    - All observations with an absolute numerical value < threshold get the class label -1 (“inactive”)

This results in a classified activity value table **T4c** as exemplified below.

classification_task_id	descriptor_id	fold_id	*input_assay_id*	*standard_qualifier*	*standard_value*	*threshold*	class_label
1	1	0	*3855277*	=	*9.38*	*9.00*	1
1	2	0	*3855277*	=	*9.58*	*9.00*	1
1	3	0	*3855277*	=	*8.56*	*9.00*	-1
1	5	0	*3855277*	=	*9.10*	*9.00*	1
1	6	0	*3855277*	=	*9.24*	*9.00*	1
1	7	0	*3855277*	=	*8.26*	*9.00*	-1
1	8	2	*3855277*	=	*8.19*	*9.00*	-1
2	1	0	*3855277*	=	*9.38*	*8.50*	1
2	2	0	*3855277*	=	*9.58*	*8.50*	1
2	3	0	*3855277*	=	*8.56*	*8.50*	1
2	5	0	*3855277*	=	*9.10*	*8.50*	1
2	6	0	*3855277*	=	*9.24*	*8.50*	1
2	7	0	*3855277*	=	*8.26*	*8.50*	-1
2	8	2	*3855277*	=	*8.19*	*8.50*	-1
3	1	0	*3855277*	=	*9.38*	*9.50*	-1
3	2	0	*3855277*	=	*9.58*	*9.50*	1
3	3	0	*3855277*	=	*8.56*	*9.50*	-1
3	5	0	*3855277*	=	*9.10*	*9.50*	-1
3	6	0	*3855277*	=	*9.24*	*9.50*	-1
3	7	0	*3855277*	=	*8.26*	*9.50*	-1

classification_task_id	descriptor_id	fold_id	*input_assay_id*	*standard_qualifier*	*standard_value*	*threshold*	class_label
3	8	2	*3855277*	=	*8.19*	*9.50*	-1
4	1	0	*3855278*	=	*6.65*	*6.00*	1
4	2	0	*3855278*	=	*6.46*	*6.00*	1
4	3	0	*3855278*	=	*6.93*	*6.00*	1
4	5	0	*3855278*	=	*6.28*	*6.00*	1
4	6	0	*3855278*	=	*6.48*	*6.00*	1
4	7	0	*3855278*	=	*6.29*	*6.00*	1
4	8	2	*3855278*	<	*6.31*	*6.00*	*
5	1	0	*3855278*	=	*6.65*	*6.50*	1
5	2	0	*3855278*	=	*6.46*	*6.50*	-1
5	3	0	*3855278*	=	*6.93*	*6.50*	1
5	5	0	*3855278*	=	*6.28*	*6.50*	-1
5	6	0	*3855278*	=	*6.48*	*6.50*	-1
5	7	0	*3855278*	=	*6.29*	*6.50*	-1
5	8	2	*3855278*	<	*6.31*	*6.50*	-1
6	1	0	*3855279*	=	*6.60*	*5.00*	1
6	2	0	*3855279*	>	*7.00*	*5.00*	1
6	3	0	*3855279*	=	*6.95*	*5.00*	1
6	5	0	*3855279*	=	*6.70*	*5.00*	1
6	6	0	*3855279*	>	*7.00*	*5.00*	1
6	7	0	*3855279*	<	*5.54*	*5.00*	1
6	8	2	*3855279*	=	*6.17*	*5.00*	1
7	1	0	*3855279*	=	*6.60*	*6.00*	1
7	2	0	*3855279*	>	*7.00*	*6.00*	1
7	3	0	*3855279*	=	*6.95*	*6.00*	1
7	5	0	*3855279*	=	*6.70*	*6.00*	1
7	6	0	*3855279*	>	*7.00*	*6.00*	1
7	7	0	*3855279*	=	*5.54*	*6.00*	-1
7	8	2	*3855279*	=	*6.17*	*6.00*	1
8	1	0	*3855279*	=	*6.60*	*7.00*	-1
8	2	0	*3855279*	>	*7.00*	*7.00*	1
8	3	0	*3855279*	=	*6.95*	*7.00*	-1
8	5	0	*3855279*	=	*6.70*	*7.00*	-1
8	6	0	*3855279*	>	*7.00*	*7.00*	1
8	7	0	*3855279*	=	*5.54*	*7.00*	-1
8	8	2	*3855279*	=	*6.17*	*7.00*	-1
9	1	0	*3855279*	=	*6.60*	*4.50*	1
9	2	0	*3855279*	>	*7.00*	*4.50*	1
9	3	0	*3855279*	=	*6.95*	*4.50*	1
9	5	0	*3855279*	=	*6.70*	*4.50*	1
9	6	0	*3855279*	>	*7.00*	*4.50*	1
9	7	0	*3855279*	=	*5.54*	*4.50*	1
9	8	2	*3855279*	=	*6.17*	*4.50*	1
10	1	0	*3855279*	=	*6.60*	*7.50*	-1
10	2	0	*3855279*	>	*7.00*	*7.50*	1
10	3	0	*3855279*	=	*6.95*	*7.50*	-1
10	5	0	*3855279*	=	*6.70*	*7.50*	-1
10	6	0	*3855279*	>	*7.00*	*7.50*	1
10	7	0	*3855279*	=	*5.54*	*7.50*	-1
10	8	2	*3855279*	=	*6.17*	*7.50*	-1
11	1	0	*3855298*	=	*1.67*	*2.00*	-1

classification_task_id	descriptor_id	fold_id	*input_assay_id*	*standard_qualifier*	*standard_value*	*threshold*	class_label
11	4	0	*3855298*	=	*2.55*	*2.00*	1
12	1	0	*9999998*	=	*1*		1
12	3	0	*9999998*	=	*-1*		-1
12	5	0	*9999998*	=	*1*		1
12	7	0	*9999998*	=	*-1*		-1
12	8	2	*9999998*	=	*-1*		-1
13	1	0	*9999999*	=	*-4.18*	*3.00*	1
13	2	0	*9999999*	=	*2.67*	*3.00*	-1
13	3	0	*9999999*	=	*1.68*	*3.00*	-1
13	5	0	*9999999*	=	*2.89*	*3.00*	-1
13	6	0	*9999999*	=	*0.10*	*3.00*	-1
13	7	0	*9999999*	=	*2.26*	*3.00*	-1
13	8	2	*9999999*	=	*-0.88*	*3.00*	-1

*Table 16: Example for a T4c table with classified activity data. Columns in italic are intermediary results for illustration purposes only and are not part of the actual table. * This row could not be classified unambiguously and is therefore removed*

4.2.5 Classification Task Filtering and Weighting

The distribution of class labels for each task is analysed across the whole dataset and per fold. Based on this, the training and evaluation data volume quorum is evaluated for each classification task. The quorum limits are read from the configuration file. The outcome is recorded in the columns **training_quorum_OK** and **evaluation_quorum_OK** which are added to the **T3c** table.

The data belonging to tasks that do not pass the training quorum are removed from the **T4c** table.

For assays of the type CATALOG-PANEL at this stage the agreed upon data volume quorum for catalogue assays is evaluated. If an assay does not meet the training quorum, then the assay type is switched to NON-CATALOG-PANEL, and the training quorum for this assay type is evaluated. If the task does not pass here, it is removed. If the assay passes as NON-CATALOG-PANEL assay it is retained as such, but the catalog_id and catalog_task_id reference is removed.

The number *n* of classifications tasks per **input_assay_id** that pass the training quorum is determined. A column **weight** is added to the T3c table. It contains, for each task passing the training quorum, the value initial_weight/n, where n represents the number of tasks passing the training quorum, and intial_weight is per default 1.0. If, for example, two task for an assay pass the training quorum, then each of the tasks will have weight 0.5. For tasks that are of the type AUX_HTS or AUX_PL the weight will be multiplied with a down weighting factor specified in the parameter file.

In addition, a column aggregation_weight is added to the **T3c** table that will be used to determine the weight of the task in the calculation of the aggregated performance metric. This column will be populated according to these rules:

- If column **is_auxiliary** is True aggregation_weight becomes 0.0
- If column **evaluation_quorum_OK** is False aggregation weight becomes 0
- In all other cases an aggregation_weight of 1.0 is used

4.2.6 Regression Task filtering and weighting

The total number of observations and the number of uncensored observations for each regression task are determined both in the whole dataset as well as per fold. Based on this the training and evaluation data volume quorum is evaluated for each regression task. The quorum limits are read from the parameter file. The outcome is recorded in the columns **training_quorum_OK** and **evaluation_quorum_OK** which are added to the **T3r** table.

Since catalogue fusion is not applied to regression, tasks of the type CATALOG-PANEL will for the regression arm be treated as NON-CATALOG-PANEL and be evaluated according to the data quorum rules for that assay type.

The data belonging to tasks fulfilling at least one of the criteria below is removed from the **T4r** table:

1. Tasks with **training_quorum_OK** = False
2. Tasks with **use_in_regression** = False

In addition, a column aggregation_weight is added to the **T3r** table that will be used to determine the weight of the task in the calculation of the aggregated performance metric. This column will be populated according to these rules:

- If column **is_auxiliary** is True aggregation_weight becomes 0.0
- If column **evaluation_quorum_OK** is False aggregation weight becomes 0
- In all other cases an aggregation_weight of 1.0 is used

A column **weight** is added to T3r and initialized with 1.0 for assays, except auxiliary data, where the auxiliary task weight from the configuration file is used. (Remark: auxiliary data is not used in regression for this year, but this is done already in preparation for later usage and to support the single partner studies preparing for this).

A column **censored_weight** is added in T3r which is the weight used for all censored datapoints of a task. This weight will be initialized based on the fraction of censored datapoints.

An example is shown in Table 18: Example for a T8r table. The toyset is too small to evaluate quorums realistically, so this part is fictitious and for illustration purposes only. The table may also include the count of observations and the count of uncensored observations in total and per fold. A copy of this table where only the rows and columns marked in blue are retained forms the T9r table, which is passed to SparseChem. Please note the conversion of the task derived from input_assay_id .

4.2.7 Filtering of descriptor data

As a consequence of the removal of rows in both **T4r** and **T4c** as described above, there might be rows in the descriptor table **T6** for which neither activity data in **T4r** nor in **T4c** is present. These rows are now identified and removed.

4.3 Assignment of continuous indexes

Background

The federated machine learning software expects the X matrix consisting of compound descriptors and the Y matrix of the assay values to be represented as a column sparse row matrix (scipy csr matrix). While the data

*processing so far has been working with data frames allowing arbitrary column and row indexes now the transition to the matrix with continuous integer indexes needs to be made. The **X** and the $\boldsymbol{Y_{reg}}$ as well as the $\boldsymbol{Y_{class}}$ must have an identical number of rows, with matching row indexes. Likewise, the indexes of **fold** numpy array must match the row indexes of **X**. The number of columns in **X** corresponds to the bitsize of the fingerprints and must be consistently assigned across all partners. Likewise, the rows of the task weight ($\boldsymbol{w_{reg}}$ and $\boldsymbol{w_{class}}$) file need to correspond to the column indices of $\boldsymbol{Y_{reg}}$ and $\boldsymbol{Y_{class}}$, respectively.*

*There will be cases where a descriptor vector has either only classification task data or regression task data. In this case either **Yreg** or **Yclass** will have an empty row.*

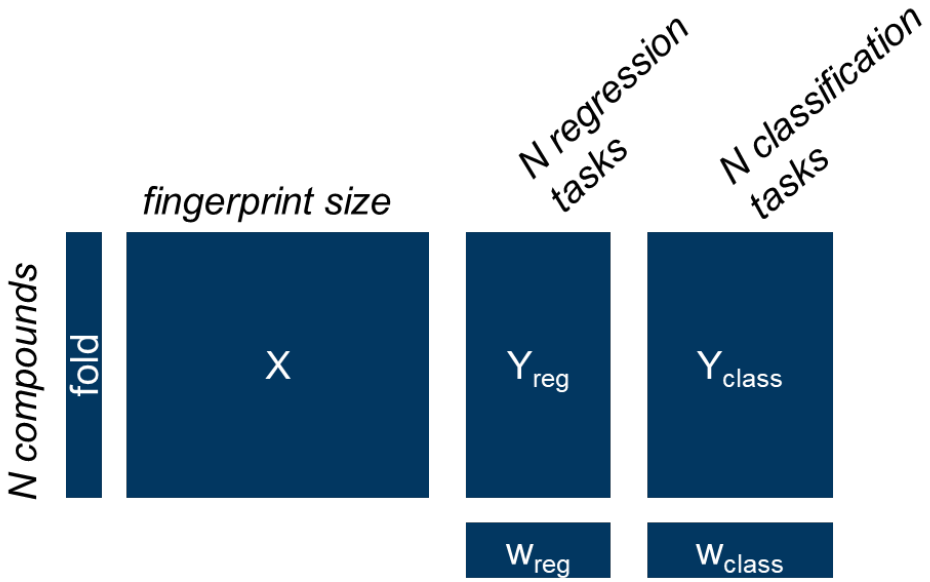

Figure 5: Input data for SparseChem. The type array used in performance evaluation.

4.3.1 Reindexing of tasks

To all entries in **T3c** which have met the training quorum, a unique **cont_classification_task_id** is assigned as a continuous range of integers starting at 0. This results in table **T8c**, which is written out. This **T8c** table still contains all the meta data carried through from the **T0** table and is thus the reference for each partner to decode the prediction results. An example is shown in Table 17. A copy of **T8c** without the metadata that are not needed during ML and without all rows not having a **cont_classification_task_id** is written out as **T9c.**

To all entries in **T3r**, which have met the training quorum and are not marked with **use_in_regression** = False, a unique **cont_regression_task_id** is assigned as a continuous range of integers starting from 0. This results in table **T8r**, which is written out. This **T8r** table still contains all the meta data carried through from the **T0** table and is thus the reference for each partner to decode the prediction results. An example is shown in Table 18.

classification_task_id	cont_classification_task_id	input_assay_id	assay_type	use_ln_regression	is_auxiliary	threshold	threshold_method	direction	training_quorum_OK	evaluation_quorum_OK	weight	aggregation_weight	Catalog_assay_id	Catal_task
1	1	3855277	OTHER		False	9.0	fixed_apaptive		True	True	0.33	1.0		
2	2	3855277	OTHER		True	8.5	aux_low		True	True	0.33	0.0		
3	3	3855277	OTHER		True	9.5	aux_high		True	False	0.33	0.0		
4	4	3855278	OTHER		False	6.0	fixed_apaptive		True	True	1.0	1.0		
5		3855278	OTHER		True	6.5	aux_high		False	False				
6	5	3855279	CATALOG-PANEL		False	5.0	expert		True	True	0.2	1.0	12	
7	6	3855279	CATALOG-PANEL		False	6.0	expert		True	True	0.2	1.0	12	
8	7	3855279	CATALOG-PANEL		False	7.0	expert		True	True	0.2	1.0	12	
9	8	3855279	CATALOG-PANEL		True	4.5	aux_low		True	True	0.2	1.0	12	
10	9	3855279	CATALOG-PANEL		True	6.5	aux_high		True	False	0.2	0.0	12	
11		3855298	ADME		False	2.0	expert	high	False	False				
12	10	9999998	AUX_PL		True		fixed		True	True	0.1	0.0		
13	11	9999999	AUX_HTS		True	3.0	fixed		True	True	0.1	0.0		

Table 17: Example for a T8c table. The toyset is too small to evaluate quorums realistically, so this part is fictitious and for illustration purposes only. The table may also include the counts for active (+1) and inactive (-1) per task in the whole set as well as per fold. A copy of this table where only the rows and columns marked in blue are retained forms the T9c table, which is passed to SparseChem.

input_ assay_id	cont_ regression_ task_id	assay_ type	use_ ln_ regression	is_ auxiliary	direction	Training_ quorum_ OK	Evaluation_ quorum_OK	weight	aggregation_ weight	fraction_ censored	censored weight
3855277	1	OTHER	True	False		True	True	1.0	1.0	0.3	0
3855278	2	OTHER	True	False		True	True	1.0	1.0	0.6	0
3855279	3	NON-CATALOG-PANEL	True	False		True	False	1.0	0.0	0.8	0
3855298		ADME	True	False	high	False	False				

Table 18: Example for a T8r table. The toyset is too small to evaluate quorums realistically, so this part is fictitious and for illustration purposes only. The table may also include the count of observations and the count of uncensored observations in total and per fold. A copy of this table where only the rows and columns marked in blue are retained forms the T9r table, which is passed to SparseChem. Please note the conversion of the task derived from input_assay_id . 3855279 to a NON-CATALOG-PANEL task.

4.3.2 Reindexing of descriptors

Due to the drop of activity data because of lacking data volume, there may now be descriptor vectors in **T6** for which no task data exists. In the **T6** table, all rows are identified which have at least one observation left in the filtered **T4r** or **T4c** table. To these rows a continuous unique **cont_descriptor_id** as integers starting from 0 is added. The **T6** table augmented in this way is written out. A copy of **T6** without the rows not having a **cont_descriptor_id** and without the **descriptor_id** column is written is written out as **T11.**

descriptor_id	cont_descriptor_id	fold_id	fp_on_bits
1	1	0	[754, 841, 926, 1039, 1515, 1673, 1787, 1804, 2000, 2226, 2947, 3534, 3881, 4444, 4708, 5247, 5548, 5969, 6009, 6695, 7326, 8094, 8272, 8428, 8438, 8648, 8737, 8833, 9231, 9383, 9639, 9768, 10507, 10819, 11361, 11605, 11607, 12122, 12260, 12418, 12559, 13043, 14721, 14723, 14847, 15498, 15560, 15663, 15880, 15984, 16164, 16278, 16756, 17044, 17083, 17293, 17552, 17850, 18297, 18304, 19569, 20206, 20329, 20410, 20528, 21099, 22068, 22156, 22789, 23169, 23394, 23469, 24273, 24326, 24351, 26070, 26490, 26624, 26919, 27003, 27066, 27207, 27645, 28908, 29119, 29303, 29539, 29609, 29810, 29955, 30140, 30456, 31097, 31112, 31795]
2	2	0	[841, 870, 906, 926, 1515, 1673, 1787, 1791, 1804, 2000, 2640, 3881, 3894, 4708, 5247, 5387, 5778, 5969, 6009, 7248, 7326, 8094, 8272, 8610, 8648, 8737, 8871, 9064, 9353, 9383, 9450, 9639, 10182, 11605, 11607, 12122, 12260, 12418, 12962, 13043, 14561, 14567, 14721, 14723, 14847, 14897, 14958, 15232, 15498, 15663, 15880, 16164, 16756, 17044, 17293, 18297, 20410, 22068, 22156, 22399, 22641, 22789, 23169, 23394, 23469, 23920, 24200, 24351, 26070, 26490, 26582, 26624, 27003, 27158, 27170, 27207, 27645, 28251, 28908, 29023, 29303, 29539, 29605, 29609, 29810, 30456, 31097, 31112, 31453]
3	3	0	*<on bit list as json string>*
4		0	*<on bit list as json string>*
5	4	0	*<on bit list as json string>*
6	5	0	*<on bit list as json string>*
7	6	0	*<on bit list as json string>*
8	7	2	*<on bit list as json string>*

Table 19: Example of a T6 table with added cont_descriptor_id. In the example case here, descriptor_id 4 is assumed to have no data left any more, as the only assay this descriptor had data for was removed because of the training quorum in both regression and classification data set. Consequently, no cont_descriptor_id is assigned to this row. The final T11 table contains only the rows and columns highlighted in blue.

4.3.3 Reindexing of activity data

The mapping of **classification_task_id** to **cont_classifcation_task_id** in **T8c** is joined to the data table **T4c** using **classification_task_id** as the join key. Likewise, the mapping from **descriptor_id** to **cont_descriptor_id** from table **T6** is joined using **descriptor_id** as the join key. The ID columns **classification_task_id** and **descriptor_id** are now dropped to obtain table **T10c.** An example can be found in Table 20.

classification_ task_id	*descriptor_id*	cont_classification_task_id	cont_descriptor_id	class_ label
1	*1*	1	1	1
1	*2*	1	2	1
1	*3*	1	3	-1
1	*5*	1	4	1
1	*6*	1	5	1
1	*7*	1	6	-1
1	*8*	1	7	-1
2	*1*	2	1	1
2	*2*	2	2	1
2	*3*	2	3	1
2	*5*	2	4	1
2	*6*	2	5	1
2	*7*	2	6	-1
2	*8*	2	7	-1
3	*1*	3	1	-1
3	*2*	3	2	1
3	*3*	3	3	-1
3	*5*	3	4	-1
3	*6*	3	5	-1
3	*7*	3	6	-1
3	*8*	3	7	-1
4	*1*	4	1	1
4	*2*	4	2	1
4	*3*	4	3	1
4	*5*	4	4	1
4	*6*	4	5	1
4	*7*	4	6	1
6	*1*	5	1	1
6	*2*	5	2	1
6	*3*	5	3	1
6	*5*	5	4	1
6	*6*	5	5	1
6	*7*	5	6	1
6	*8*	5	7	1
7	*1*	6	1	1
7	*2*	6	2	1
7	*3*	6	3	1
7	*5*	6	4	1
7	*6*	6	5	1
7	*7*	6	6	-1
7	*8*	6	7	1
8	*1*	7	1	-1
8	*2*	7	2	1
8	*3*	7	3	-1
8	*5*	7	4	-1

classification_task_id	*descriptor_id*	cont_classification_task_id	cont_descriptor_id	class_label
8	*6*	7	5	1
8	*7*	7	6	-1
8	*8*	7	7	-1
9	*1*	8	1	1
9	*2*	8	2	1
9	*3*	8	3	1
9	*5*	8	4	1
9	*6*	8	5	1
9	*7*	8	6	1
9	*8*	8	7	1
10	*1*	9	1	-1
10	*2*	9	2	1
10	*3*	9	3	-1
10	*5*	9	4	-1
10	*6*	9	5	1
10	*7*	9	6	-1
10	*8*	9	7	-1
12	*1*	10	1	1
12	*3*	10	3	-1
12	*5*	10	4	1
12	*7*	10	6	-1
12	*8*	10	7	-1
13	*1*	11	1	1
13	*2*	11	2	-1
13	*3*	11	3	-1
13	*5*	11	4	-1
13	*6*	11	5	-1
13	*7*	11	6	-1
13	*8*	11	7	-1

Table 20: Example of a T10c table. The columns in italic are intermediate results shown for illustration purposes, they are not kept in the final T10c table.

The mapping of **input_assay_id** to **cont_regression_task_id** in **T8r** is joined to the data table **T4r** using **input_assay_id** as the join key. Likewise, the mapping from **descriptor_id** to **cont_descriptor_id** from table **T6** is joined using **descriptor_id** as the join key. The ID columns **input_assay_id** and **descriptor_id** are now dropped to obtain table **T10r.** An example can be found in Table 21.

input_assay_id	*descriptor_id*	cont_regression_task_id	cont_descriptor_id	standard_qualifier	standard_value
3855277	*1*	1	1	=	9.38
3855277	*2*	1	2	=	9.58
3855277	*3*	1	3	=	8.56
3855277	*5*	1	4	=	9.10
3855277	*6*	1	5	=	9.24
3855277	*7*	1	6	=	8.26
3855277	*8*	1	7	=	8.19
3855278	*1*	2	1	=	6.65
3855278	*2*	2	2	=	6.46
3855278	*3*	2	3	=	6.93

3855278	*5*	2	4	=	6.28
3855278	*6*	2	5	=	6.48
3855278	*7*	2	6	=	6.29
3855278	*8*	2	7	=	6.31
3855279	*1*	3	1	=	6.60
3855279	*2*	3	2	>	7.00
3855279	*3*	3	3	=	6.95
3855279	*5*	3	4	=	6.70
3855279	*6*	3	5	>	7.00
3855279	*7*	3	6	=	5.54
3855279	*8*	3	7	=	6.17

Table 21: Example of a T10r table. Columns in italics are only intermediate results and not kept as part of this table.

4.3.4 Translation into numpy matrices and arrays

Data table **T11** is used to construct the **X** scipy csr matrix with **cont_descriptor_vector_id** as row index and the already permuted fingerprint bit index as column index. The numpy fold array is extracted from the **fold_id** column of **T11.**

From **T10c** the $\mathbf{Y_{class}}$ scipy csr matrix is constructed using **cont_descriptor_vector_id** as row index, **cont_classification_task_id** as column index, and the **class_label** as values. $\mathbf{Y_{class}}$ is of the integer datatype and missing values are represented as 0.

From **T10r** the $\mathbf{Y_{reg}}$ scipy csr matrix is constructed, using again **cont_descriptor_vector_id** as row index, the **cont_regression_task_id** as column index, and the **standard_value** as values. $\mathbf{Y_{reg}}$ is of data type float. In the same way the $\mathbf{Y_{censor}}$ csr matrix is constructed this time with **standard_qualifier** encoded as integer values. Equality (=), meaning no censoring is encoded as 0, > is encoded as +1.0, and < is encoded as -1.

T9c and **T9r** are passed to SparseChem as csv files directly.

5 Public Data

In general, the preparation of public data follows the same principles as for the private pharma data. Unless stated explicitly below, the same process will be used. However, the different inherent structure of the data requires some deviations from this general principle.

ChEMBL[17,18] will be the only public data source used for concentration response data. Like the pharma data, the ChEMBL data is limited to concentration-response experiments and added ADME assays. Assays for which at least part of the data points have pChEMBL values are considered as concentration-response assays. This does, however, not mean that from these assays only data points with a pChEMBL value are retained. Compounds with an activity value having qualifiers (such as >), which do not have a pChEMBL value, will be kept as well in accordance with the rules applied by pharma (see § 2.1.1).

One of the differences between the ChEMBL and the pharma data is the disproportional high number of assays in ChEMBL (1.1 M assay before applying any data volume quorum). Only a fraction of those assays has sufficient data to be included. Where both the assay metadata, and the activity value distribution suggest it makes sense to combine several closely analogous assays for the same target in one prediction endpoint, this may have been done.

Auxiliary data from public sources is not used.

Iktos will prepare the public data until the MELLODDY Tuner ready stage (T0, T1, and T2) and will hand them over to Servier for preparation with MELLODDY Tuner and hosting on the federated platform.

6 Staging of the Data

6.1 Verification of Correct Setup

It is crucial that all partners use the same setup especially for calculating and scrambling the chemical fingerprints. Notably, the choice of a wrong scrambling key will lead to a silent failure of the federated run, as it is not detectable by another partner that a wrong setup is used. In order to reduce the risk of an erroneous key being used, the following strategy is applied:

1. All parameters influencing the content of the output, with exception of the secret key, are grouped together in one parameter file and is distributed together with the MELLODDY-Tuner code. This will reduce the risk for copy and paste errors.

2. The secret key needs to be distributed by secured channels to pharma partners only as it contains the private permutation key.

3. The MELLODDY-Tuner code has an embedded small reference data set consisting of public chemical structures. At every execution of the MELLODDY-Tuner code, also this dataset is processed. A checksum is generated from the resulting reference output. All partners will compare the checksum generated by their pre-processing run with the checksum communicated for the correct setup, code release, and the correct secret key. This will protect against using a wrong setup file.

4. Unit tests will be performed by each GitLab commit to the “develop” branch. For each new functionality an appropriate unit test is highly recommended.

5. The release mechanism in GitLab is used to produce well defined releases.

6.2 Validation with Sparsechem Runs

6.2.1 Verification of machine learning outcome

The purpose of this is to assert that the data set is fit for learning. The following procedure is recommended: a model is trained on the year 3 data as generated by MELLODDY-Tuner, using the optimal hyper-parameters from year 2. The overall performance is compared and should not be substantially lower than year 2, given that the fold splitting hasn’t been changed. If a lower overall performance has been detected, then a task by task comparison should be done, to ensure that at least those tasks that were present in year 2 did not perform substantially worse in year 3 compared to the year 2 results.

6.2.2 Assessment of the GPU memory footprint

The purpose of this check is to anticipate the memory footprint of the dataset in the federated run, especially the GPU memory footprint. In order to do this a Sparsechem version allowing determining the peak memory usage is available. Using this the GPU memory usage can be assessed for a given dataset in combination with the network architecture and size, and internal batch size used. For each of the data sets used, the memory consumption will be determined by running a model for 2 epochs each, using an architecture that corresponds to the requirements of the federated run.

6.3 Cloud upload and asset registration

The lineage of the data sets that are considered for year 3 is outlined in Figure 6 below. Two MELLODDY-Tuner runs are required, one without auxiliary data and one with auxiliary data. Each of these runs produces the output for the three different modelling modalities classification (**cls**), classification-regression hybrid (**hybrid**), and regression (**reg**), in a separate subfolder per modality. As there are no auxiliary regression tasks, the content of with-aux-reg equals the content of no-aux-reg and is therefore not needed, this leaves the data sets **no-aux-cls**, **no-aux-hybrid**, **no-aux-reg**, **with-aux-cls**, and **with-aux-hybrid**.

Once the platform is cleared for dataset upload, the operational contacts will upload each of these 5 datasets onto the platform. For each dataset there will be three clones created into a subfolder each for the three model building phases phase1 (hyper-parameter tuning), phase 2 (performance evaluation on independent test set), phase 3 (model trained with all available data). These folders are then used in the data registration, upon instruction by the run coordinators from WP6.

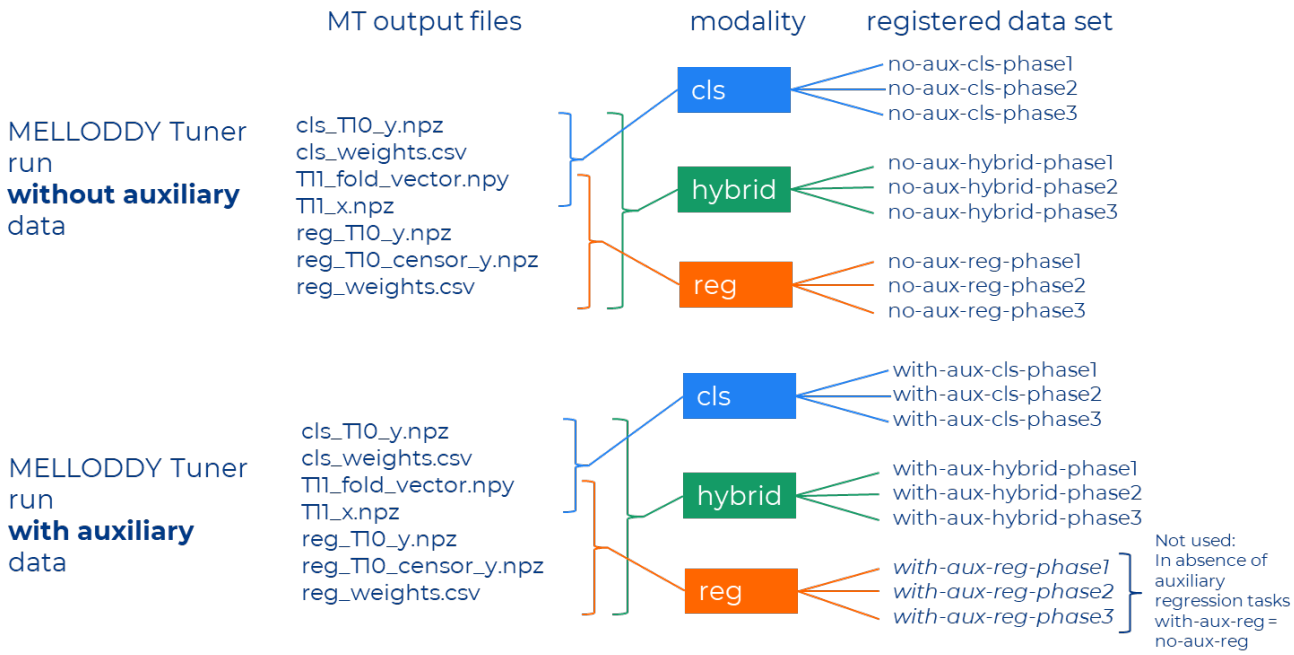

Figure 6: Lineage of the datasets to be considered for year 3.

7 References

(1) Seidler, J.; McGovern, S. L.; Doman, T. N.; Shoichet, B. K. Identification and Prediction of Promiscuous Aggregating Inhibitors among Known Drugs. *J. Med. Chem.* **2003**. https://doi.org/10.1021/jm030191r.

(2) Rana, P.; Naven, R.; Narayanan, A.; Will, Y.; Jones, L. H. Chemical Motifs That Redox Cycle and Their Associated Toxicity. https://doi.org/10.1039/c3md00149k.

(3) Hermann, J. C.; Chen, Y.; Wartchow, C.; Menke, J.; Gao, L.; Gleason, S. K.; Haynes, N. E.; Scott, N.; Petersen, A.; Gabriel, S.; Vu, B.; George, K. M.; Narayanan, A.; Li, S. H.; Qian, H.; Beatini, N.; Niu, L.;

Gan, Q. F. Metal Impurities Cause False Positives in High-Throughput Screening Campaigns. *ACS Med. Chem. Lett.* **2013**, *4* (2), 197–200. https://doi.org/10.1021/ml3003296.

(4) Stock, U.; Matter, H.; Diekert, K.; Dörner, W.; Dröse, S.; Licher, T. Measuring Interference of Drug-Like Molecules with the Respiratory Chain: Toward the Early Identification of Mitochondrial Uncouplers in Lead Finding. *Assay Drug Dev. Technol.* **2013**, *11* (7), 408–422. https://doi.org/10.1089/adt.2012.463.

(5) Beck, B. BioProfile - Extract Knowledge from Corporate Databases to Assess Cross-Reactivities of Compounds. *Bioorganic Med. Chem.* **2012**, *20* (18), 5428–5435. https://doi.org/10.1016/j.bmc.2012.04.023.

(6) Baell, J. B.; Holloway, G. A. New Substructure Filters for Removal of Pan Assay Interference Compounds (PAINS) from Screening Libraries and for Their Exclusion in Bioassays. *J. Med. Chem.* **2010**, *53* (7), 2719–2740. https://doi.org/10.1021/jm901137j.

(7) Bowes, J.; Brown, A. J.; Hamon, J.; Jarolimek, W.; Sridhar, A.; Waldron, G.; Whitebread, S. Reducing Safety-Related Drug Attrition: The Use of in Vitro Pharmacological Profiling. *Nat. Rev. Drug Discov.* **2012**, *11* (12), 909–922. https://doi.org/10.1038/nrd3845.

(8) Zhu, X. W.; Sedykh, A.; Zhu, H.; Liu, S. S.; Tropsha, A. The Use of Pseudo-Equilibrium Constant Affords Improved QSAR Models of Human Plasma Protein Binding. *Pharm. Res.* **2013**, *30* (7), 1790–1798. https://doi.org/10.1007/s11095-013-1023-6.

(9) Gubler, H. Methods for Statistical Analysis, Quality Assurance and Management of Primary High-throughput Screening Data. In *Methods and Principles in Medicinal Chemistry*; Wiley-VCH, 2006; Vol. 28, pp 151–205. https://doi.org/10.1002/9783527609321.ch7.

(10) M Nissink, J. W.; Blackburn, S. Quantification of Frequent-Hitter Behavior Based on Historical High-Throughput Screening Data. *Future Med. Chem.* **2014**, *6* (10), 1113–1126. https://doi.org/10.4155/fmc.14.72.

(11) Sitzmann, M.; Ihlenfeldt, W. D.; Nicklaus, M. C. Tautomerism in Large Databases. *J. Comput. Aided. Mol. Des.* **2010**, *24* (6–7), 521–551. https://doi.org/10.1007/s10822-010-9346-4.

(12) Bemis, G. W.; Murcko, M. A. The Properties of Known Drugs. 1. Molecular Frameworks. *J. Med. Chem.* **1996**, *39* (15), 2887–2893. https://doi.org/10.1021/jm9602928.

(13) Kruger, F.; Stiefl, N.; Landrum, G. A. RdScaffoldNetwork: The Scaffold Network Implementation in RDKit. *J. Chem. Inf. Model.* **2020**, *60* (7), 3331–3335. https://doi.org/10.1021/acs.jcim.0c00296.

(14) Schuffenhauer, A.; Ertl, P.; Roggo, S.; Wetzel, S.; Koch, M. A.; Waldmann, H. The Scaffold Tree − Visualization of the Scaffold Universe by Hierarchical Scaffold Classification. *J. Chem. Inf. Model.* **2007**, *47* (1), 47–58. https://doi.org/10.1021/ci600338x.

(15) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50* (5), 742–754. https://doi.org/10.1021/ci100050t.

(16) Xia, X.; Maliski, E. G.; Gallant, P.; Rogers, D. Classification of Kinase Inhibitors Using a Bayesian Model. *J. Med. Chem.* **2004**, *47* (18), 4463–4470. https://doi.org/10.1021/jm0303195.

(17) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: A Large-Scale Bioactivity Database for Drug Discovery. *Nucleic Acids Res.* **2012**, *40* (D1). https://doi.org/10.1093/nar/gkr777.

(18) Gaulton, A.; Hersey, A.; Nowotka, M.; Bento, A. P.; Chambers, J.; Mendez, D.; Mutowo, P.; Atkinson, F.; Bellis, L. J.; Cibrián-Uhalte, E.; Davies, M.; Dedman, N.; Karlsson, A.; Magariños, M. P.; Overington, J. P.; Papadatos, G.; Smit, I.; Leach, A. R. The ChEMBL Database in 2017. *Nucleic Acids Res.* **2017**, *45* (D1), D945–D954. https://doi.org/10.1093/nar/gkw1074.

Annexes

Annex 1

input_compound_id	input_assay_id	*Input activity data*				standard_qualifier / standard_value	
3950540	3855277	*Ki*	=	*0.26*	*nM*	=	9.58
3955505	3855277	*Ki*	=	*0.42*	*nM*	=	9.38
3896638	3855277	*Ki*	=	*0.58*	*nM*	=	9.24
3946901	3855277	*Ki*	=	*0.79*	*nM*	=	9.10
3927895	3855277	*Ki*	=	*1.05*	*nM*	=	8.98
3984387	3855277	*Ki*	=	*2.75*	*nM*	=	8.56
3905582	3855277	*Ki*	=	*3.16*	*nM*	=	8.50
3935776	3855277	*Ki*	=	*3.24*	*nM*	=	8.49
3983937	3855277	*Ki*	=	*5.50*	*nM*	=	8.26
3901587	3855277	*Ki*	=	*6.17*	*nM*	=	8.21
85606	3855277	*Ki*	=	*6.46*	*nM*	=	8.19
3967683	3855277	*Ki*	=	*7.76*	*nM*	=	8.11
3935776	3855277	*Ki*	=	*9.33*	*nM*	=	8.03
3913062	3855277	*Ki*	=	*17.78*	*nM*	=	7.75
3958494	3855277	*Ki*	=	*25.12*	*nM*	=	7.60
3958494	3855277	*Ki*	=	*28.18*	*nM*	=	7.55
3901211	3855277	*Ki*	=	*41.69*	*nM*	=	7.38
3927895	3855277	*Ki*	=	*72.44*	*nM*	=	7.14
3913062	3855277	*Ki*	=	*794.33*	*nM*	=	6.10
3984387	3855278	*Ki*	=	*117.49*	*nM*	=	6.93
3905582	3855278	*Ki*	=	*128.82*	*nM*	=	6.89
3955505	3855278	*Ki*	=	*223.87*	*nM*	=	6.65
3935776	3855278	*Ki*	=	*309.03*	*nM*	=	6.51
3935776	3855278	*Ki*	=	*316.23*	*nM*	=	6.50
3896638	3855278	*Ki*	=	*331.13*	*nM*	=	6.48
3927895	3855278	*Ki*	=	*331.13*	*nM*	=	6.48
3950540	3855278	*Ki*	=	*346.74*	*nM*	=	6.46
85606	3855278	*Ki*	>	*489.78*	*nM*	<	6.31
3958494	3855278	*Ki*	=	*512.86*	*nM*	=	6.29
3946901	3855278	*Ki*	=	*524.81*	*nM*	=	6.28
3967683	3855278	*Ki*	=	*630.96*	*nM*	=	6.20
3901211	3855278	*Ki*	=	*707.95*	*nM*	=	6.15
3983937	3855278	*Ki*	=	*794.33*	*nM*	=	6.10
3913062	3855278	*Ki*	=	*933.25*	*nM*	=	6.03
3901587	3855278	*Ki*	=	*977.24*	*nM*	=	6.01
3958494	3855278	*Ki*	=	*1000.00*	*nM*	=	6.00
3927895	3855278	*Ki*	=	*4365.16*	*nM*	=	5.36
3913062	3855278	*Ki*	=	*16982.44*	*nM*	=	4.77
3950540	3855279	*IC50*	<	*100.00*	*nM*	>	7.00
3896638	3855279	*IC50*	<	*100.00*	*nM*	>	7.00
3984387	3855279	*IC50*	=	*112.20*	*nM*	=	6.95
3946901	3855279	*IC50*	=	*200.00*	*nM*	=	6.70
3946901	3855279	*IC50*	=	*208.93*	*nM*	=	6.68

input_compound_id	input_assay_id	*Input activity data*				standard_qualifier / standard_value	
3935776	3855279	*IC50*	=	*251.19*	*nM*	=	6.60
3901587	3855279	*IC50*	=	*446.68*	*nM*	=	6.35
3955505	3855279	*IC50*	=	*524.81*	*nM*	=	6.28
85606	3855279	*IC50*	=	*676.08*	*nM*	=	6.17
3927895	3855279	*IC50*	=	*691.83*	*nM*	=	6.16
3983937	3855279	*IC50*	=	*2884.03*	*nM*	=	5.54
3901211	3855279	*IC50*	=	*3630.78*	*nM*	=	5.44
3958494	3855279	*IC50*	=	*3981.07*	*nM*	=	5.40
3913062	3855279	*IC50*	=	*4677.35*	*nM*	=	5.33
3905582	3855279	*IC50*	>	*10000.00*	*nM*	<	5.00
3901587	3855298	*CLInt*	=	*20.00*	*mL.min-1.g-1*	=	1.30
3955505	3855298	*CLInt*	=	*107.00*	*mL.min-1.g-1*	=	2.03
3941818	3855298	*CLInt*	=	*352.00*	*mL.min-1.g-1*	=	2.55
3935776	9999999	*Rscore*	=	*-4.18*		=	-4.18
3958494	9999999	*Rscore*	=	*-1.60*		=	-1.60
3927895	9999999	*Rscore*	=	*-1.29*		=	-1.29
3913062	9999999	*Rscore*	=	*-0.95*		=	-0.95
85606	9999999	*Rscore*	=	*-0.88*		=	-0.88
3901587	9999999	*Rscore*	=	*-0.78*		=	-0.78
3905582	9999999	*Rscore*	=	*-0.21*		=	-0.21
3896638	9999999	*Rscore*	=	*0.10*		=	0.10
3901211	9999999	*Rscore*	=	*0.65*		=	0.65
3946901	9999999	*Rscore*	=	*0.76*		=	0.76
3955505	9999999	*Rscore*	=	*1.05*		=	1.05
3984387	9999999	*Rscore*	=	*1.68*		=	1.68
3983937	9999999	*Rscore*	=	*2.26*		=	2.26
3950540	9999999	*Rscore*	=	*2.67*		=	2.67
3967683	9999999	*Rscore*	=	*2.89*		=	2.89

Table 22: Copy of the T1 table example as in

input_compound_id	input_assay_id	standard_qualifier	standard_value
3950540	3855277	=	9.58
3955505	3855277	=	9.38
3896638	3855277	=	9.24
3946901	3855277	=	9.10
3927895	3855277	=	8.98
3984387	3855277	=	8.56
3905582	3855277	=	8.50
3935776	3855277	=	8.49
3983937	3855277	=	8.26
3901587	3855277	=	8.21
85606	3855277	=	8.19
3967683	3855277	=	8.11
3935776	3855277	=	8.03
3913062	3855277	=	7.75
3958494	3855277	=	7.60
3958494	3855277	=	7.55
3901211	3855277	=	7.38
3927895	3855277	=	7.14

input_compound_id	input_assay_id	standard_qualifier	standard_value
3913062	3855277	=	6.10
3984387	3855278	=	6.93
3905582	3855278	=	6.89
3955505	3855278	=	6.65
3935776	3855278	=	6.51
3935776	3855278	=	6.50
3896638	3855278	=	6.48
3927895	3855278	=	6.48
3950540	3855278	=	6.46
85606	3855278	<	6.31
3958494	3855278	=	6.29
3946901	3855278	=	6.28
3967683	3855278	=	6.20
3901211	3855278	=	6.15
3983937	3855278	=	6.10
3913062	3855278	=	6.03
3901587	3855278	=	6.01
3958494	3855278	=	6.00
3927895	3855278	=	5.36
3913062	3855278	=	4.77
3950540	3855279	>	7.00
3896638	3855279	>	7.00
3984387	3855279	=	6.95
3946901	3855279	=	6.70
3946901	3855279	=	6.68
3935776	3855279	=	6.60
3901587	3855279	=	6.35
3955505	3855279	=	6.28
85606	3855279	=	6.17
3927895	3855279	=	6.16
3983937	3855279	=	5.54
3901211	3855279	=	5.44
3958494	3855279	=	5.40
3913062	3855279	=	5.33
3905582	3855279	<	5.00
3901587	3855298	=	1.30
3955505	3855298	=	2.03
3941818	3855298	=	2.55
3901587	9999998	=	1
85606	9999998	=	-1
3967683	9999998	=	1
3935776	9999998	=	1
3913062	9999998	=	-1
3958494	9999998	=	-1
3905582	9999998	=	1
3901211	9999998	=	-1
3927895	9999998	=	-1
3913062	9999998	=	-1
3935776	9999999	=	-4.18
3958494	9999999	=	-1.60
3927895	9999999	=	-1.29

input_compound_id	input_assay_id	standard_qualifier	standard_value
3913062	9999999	=	-0.95
85606	9999999	=	-0.88
3901587	9999999	=	-0.78
3905582	9999999	=	-0.21
3896638	9999999	=	0.10
3901211	9999999	=	0.65
3946901	9999999	=	0.76
3955505	9999999	=	1.05
3984387	9999999	=	1.68
3983937	9999999	=	2.26
3950540	9999999	=	2.67
3967683	9999999	=	2.89

Table 7 additionally containing the original values before conversion