

Additional file 4 to: Can it be checked? A cross-sectional meta-research audit of access to, and reporting in, machine learning studies of pharmaceutical manufacturing and quality control.
Phani Kumar Balagam.

Prompts given to the language model

Model: Claude (Anthropic), configured identifier `claude-opus-5-5`. Each instance received the text below, unchanged except for the file names shown in braces.

Prompts used by the model (verbatim)

Two prompts were used, one for title-and-abstract screening and one for rubric scoring. Each was fixed before the step it governs began, and each was given unchanged to every model instance, with only the batch file name substituted where shown as `{BATCH_FILE}` / `{OUTPUT_FILE}`.

Prompt 1 - title and abstract screening

You are screening bibliographic records for a meta-research study of machine learning (ML) applied to pharmaceutical manufacturing and quality control. Read the instructions, then screen every record in `{BATCH_FILE}` (JSON Lines; fields: key, title, abstract, venue, dates, pubtypes). Write one JSON object per record to `{OUTPUT_FILE}` with fields:
key, decision ("include", "exclude" or "unclear"), reason (one code below), note (at most 15 words, optional).

INCLUDE only if ALL of these hold on the title and abstract:

I1. Primary research that fits or evaluates at least one machine-learning model on data

(supervised or unsupervised learning: neural networks of any kind, tree ensembles, random forests, gradient boosting, support vector machines or regression, k-nearest

neighbors, Gaussian process regression, decision trees, or clustering used for prediction). Classical chemometrics alone (PLS, PCA, PCR, MLR, DoE response surfaces)

does NOT qualify unless an ML model of the kind listed is also fitted.

I2. The data are generated by the MANUFACTURE or QUALITY CONTROL of a human drug product,

drug substance or biologic (small molecule, therapeutic protein, monoclonal antibody,

vaccine): process development or commercial processing, unit operations (e.g.,

granulation, blending, compression, coating, drying, crystallization, lyophilization, spray drying, extrusion, cell culture/bioreactor, purification), process analytical technology, in-process controls, release or stability testing, or analytical methods applied to drug products or substances for quality control.

I3. At least one quantitative performance result of the ML model is reported or evidently reported (e.g., accuracy, R2, RMSE, AUC).

EXCLUDE with one reason code:

E1_no_ml	no ML model fitted (review of ML, chemometrics only, mechanistic model only)
E2_domain	not pharmaceutical manufacturing or QC (e.g., food, cosmetics, chemicals, agriculture, environment, devices, education, general engineering)
E3_discovery	drug discovery, QSAR, ADMET, molecular design, target identification
E4_clinical	clinical prediction, pharmacokinetics, bioequivalence, pharmacovigilance, hospital or pharmacy practice, medication management, point-of-care printing
E5_supply	supply chain, demand forecasting, sales, commercial operations
E6_herbal	herbal, traditional-medicine or natural-product raw-material authentication
E7_falsified	falsified or counterfeit medicine detection by market-surveillance testing
E8_genetx	gene-therapy vector sequence or genome characterization
E9_property	physicochemical property prediction (e.g., solubility) without process or product quality data
E10_type	review, perspective, editorial, tutorial, protocol, preprint, thesis, book chapter, conference abstract without results
E11_no_result	ML is proposed or described but no performance result is reported
E12_other	other (explain in note)

Use "unclear" (reason U_needs_fulltext) only when the title and abstract do not settle I1, I2 or I3 and the full text would. A missing abstract is not by itself a reason to include.

Decide each record independently. Do not use any knowledge about a record beyond its fields.

Prompt 2 - rubric scoring

You are scoring published articles for a meta-research audit of how machine-learning (ML) studies in pharmaceutical manufacturing and quality control are reported. {BATCH_FILE} lists a batch of articles, each with the path of its full-text file. Score each article separately, from its own full-text file only, and write one JSON object per article, one per line, to {OUTPUT_FILE}. Never infer an item from general knowledge, from the journal, from another article, or from what authors usually do.

Step 1. Headline result. Identify the headline result: the first quantitative performance result of the proposed ML model stated in the abstract; if the abstract states none, the best model's main performance result in the Results. Record it (at most 15 words, quoted).

Step 2. Eligibility check on the full text. If the article does not meet the screening criteria I1-I3 (see the screening prompt), record eligible = false with the exclusion code, and stop.

Step 3. Score items R1-R9 for the model and data behind the headline result. Each item takes exactly one value:

Y = the full text reports it (quote the evidence),

N = the full text does not report it, or reports its absence (quote the statement if one exists; otherwise evidence = "not reported" and list the search terms used),

NA = not applicable (only where the item says so),

NASS = not assessable: the evidence is in a part of the article that could not be reached

(for example a supplementary file or a linked page that did not open). Never use NASS

for something that is simply not written in the text.

Every Y and every N-with-statement carries a locator (section heading, table or figure

number, or page) and a verbatim quote of at most 15 words.

R1 Data availability. Y if the data behind the headline result are publicly accessible: a

repository link or DOI, a supplementary data file, data printed in the article in full, or

a cited public dataset that was used as is. N if "available on request", "confidential",

"proprietary", or no statement.

R2 Code availability. Y if the analysis code is publicly accessible (repository link, DOI or supplementary code file). Naming software, libraries or versions is not code availability. N otherwise, including "on request".

R3 Data split stated. Y if the text states how data were partitioned for evaluation (hold-out proportions or counts, k-fold or leave-one-out cross-validation, split by batch, time or site). N otherwise.

R4 Held-out evaluation. Y if the headline result is computed on data not used to fit the model or to select it or its hyperparameters (an untouched test set, an external set, or nested cross-validation). N if it comes from training data, from the same cross-validation used for tuning or model selection, or if the text is silent on this.

R5 Uncertainty. Y if the headline performance metric is reported with a confidence interval, standard deviation, standard error or range over repeats. N otherwise.

R6 Baseline comparison. Y if the proposed model is compared on the same data with at least one simpler or incumbent method (a linear, PLS or other classical model, the current practice, a null or mean predictor, or a previously published method). Comparing only several complex ML models with each other is N.

R7 Class distribution. For a classification task: Y if class counts or proportions are reported for the data used; N otherwise. For regression or other non-classification tasks: NA.

R8 External validation. Y if the model is evaluated on data from a different batch campaign, scale, site, instrument, time period, or an independent dataset not drawn from the development data. A random split of one dataset is N.

R9 Repeated evaluation. Y if performance is evaluated over more than one random split, seed or resample (repeated cross-validation, bootstrap, multiple runs). A single k-fold cross-validation counts as Y only if fold-level results or their spread are reported.

Step 4. Descriptors: task (classification, regression, clustering, other); ML family (neural network, tree ensemble, SVM/SVR, Gaussian process, k-NN, other); data provenance (industrial, laboratory or pilot, public dataset, simulated, not stated).

Output one JSON object with fields: key, eligible, exclusion_code, headline, R1..R9 (each an object {value, locator, quote}), task, ml_family, provenance, notes (at most 30 words).

Operational wrapper placed before Prompt 1 (screening)

OPERATIONAL WRAPPER (plumbing only; the screening rules are the prompt below, verbatim).

Read the whole batch file with the Read tool (use offset and limit to page through it if it is long; records are separated by lines beginning "### key:"). Screen every record. Write the output file with the Write tool as JSON Lines: exactly one JSON object per record, in the same order as the batch. Then verify with Bash (wc -l) that the output line count equals the record count in the batch header, and fix it if not. Do not use the web and do not read any other file. Your final reply must be one line only: "done <records> include=<n> unclear=<n> exclude=<n>".

Operational wrapper placed before Prompt 2 (scoring)

Prompt 2 was followed by Prompt 1 verbatim, because step 2 of Prompt 2 refers to the screening criteria.

OPERATIONAL WRAPPER (plumbing only; the scoring rules are the scoring prompt below, verbatim).

The batch file has one header line, then one line per article:

key<TAB>full-text file<TAB>lines.

Take the articles one at a time, in order. For each, read its full-text file completely with the

Read tool (files are long: page through with offset and limit, 400 lines per read, until the last

line), then score it under the scoring prompt before opening the next file. Page markers look like

"[page N]" and section headings start with "## "; use them as locators. The screening criteria

I1-I3 and the exclusion codes that Step 2 refers to are in the screening prompt reproduced after

the scoring prompt; only its criteria and codes apply, not its file-handling sentences. Write the output file with the Write tool as JSON Lines: exactly one JSON

object per article, in batch order, with exactly these fields: key, eligible (true or false),

exclusion_code (a code from the screening prompt, or null), headline, R1 ... R9 (each an object

{"value": "Y"|"N"|"NA"|"NASS", "locator": string, "quote": string}), task, ml_family,

provenance, notes. For an ineligible article set R1..R9 to null. Every quote must be copied character for character from the full-text file, at most 15 words; use "" when there is none. Then verify with Bash that the file has one valid JSON object per article (for example `python3 -c "import json;[json.loads(l) for l in open('OUT')]"` and `wc -l`) and fix it if not. Do not use the web and do not read any other file. Your final reply must be one line only:
"done <articles> eligible=<n>".

Instruction for the DR-01 re-scoring of item R1 (10 papers)

OPERATIONAL WRAPPER. Re-score item R1 only, for each article listed in <scoring folder>/in/dr01_1.txt (one per line: key<TAB>full-text file<TAB>lines<TAB>the first-pass R1 quote). Read each full-text file completely with the Read tool (400 lines per read, until the last line), with particular attention to data statements, tables, appendices and supplementary-material descriptions. Score R1 under the rule below and the clarification DR-01 below. Write <scoring folder>/out/dr01_1.jsonl as JSON Lines, one object per article in order: {"key": ..., "R1": {"value": "Y"|"N"|"NASS", "locator": ..., "quote": ...}, "note": "at most 30 words"}. Quotes are copied character for character from the file, at most 15 words. Do not use the web and do not read any other file. Reply with one line: "done <n> Y=<n> N=<n> NASS=<n>".

===== R1 RULE (verbatim from the scoring prompt) =====
R1 Data availability. Y if the data behind the headline result are publicly accessible: a repository link or DOI, a supplementary data file, data printed in the article in full, or a cited public dataset that was used as is. N if "available on request", "confidential", "proprietary", or no statement.

===== CLARIFICATION DR-01 (verbatim from DECISION-RULE-LOG.md) =====
Clarification. Under R1, a data-availability statement counts as Y only when it locates the data behind the headline result:
- a named repository, DOI or URL;
- a named supplementary file that the statement or the text says holds the data; or
- a table or appendix in the article that holds the modelled observations, found in the text.

If the statement points only to supplementary files, and nothing in the text shows what they hold, the value is NASS (the evidence is in a part of the article that could not be reached). If the data are printed in the article, give the table. Otherwise the value is N.