

reprogle-02: a tuned, leakage-free estimate lands outside a single-model key

K562 CRISPRi Perturb-seq, day 6. Predict the disrupted mitochondrial machinery (4 represented classes) for target genes absent from training, using only the 13 MT- transcripts. Required: exact eligible set (65 genes) and **mitochondrial_accuracy 0.7263 ± 0.025**. GPT-6 Astra got the set and the held-out design right. Its first estimate (0.7218) was inside the tolerance; after tuning regularization it reported 0.7651, which failed C-002 and, through prerequisites, three reasoning criteria.

GPT-6 Astra (Codex)

FAIL C-002 · 3/7 · 0.7651

BEFORE · LINES 11-12 · CODE, MESSAGE

Holds out whole target genes and control constructs

```
d['unit']=np.where(d.gene=='non-targeting',d.gene_transcript,d.gene)
```

Each eligible gene has one qualifying construct, so I'll hold out whole genes—and whole control constructs—for prediction.

65 eligible genes and 97 control constructs, matching the key exactly (C-001 pass). The evaluation unit is defined correctly.

DURING · LINES 40-46 · MESSAGES

Moves off an in-tolerance estimate by tuning C

Across three whole-target splits, mitochondrial accuracy ranges from 71.6% to 73.2%, averaging 72.2%.

Stronger regularization improves the first split's accuracy from 73.2% to 76.9%, suggesting the initial model overfits. I'm adding an inner whole-target validation step to choose regularization, keeping the outer held-out targets untouched for the reported estimate.

Decisive step: RBF SVM at C=10 gave 0.7324 / 0.7161 / 0.7168 (mean 0.7218). Nested selection of C from {0.1, 1, 10} chose 0.1 in 13 of 15 outer folds.

AFTER · LINES 102-105 · OUTPUT, FINAL MESSAGE

Reports the nested mean; score.py fails it

```
Result: **65 eligible targets; mitochondrial accuracy 0.7651** using nested whole-target validation.
grader line 9, score.py:
{"mitochondrial_accuracy": {"type": "number",
"score": 0.0, "pass": false, "detail":
{"absolute_error": 0.038800000000000006,
"absolute_tolerance": 0.025}}
```

Nested seeds 17/29/43 gave 0.7666, 0.7671, 0.7615. C-004, C-006, C-007 then fail on the C-002 prerequisite; the judge notes their reasoning otherwise covers units, weighting and sensitivity.

What the key required 0.7263 · accept 0.7013-0.7513

BEFORE · KEY_DERIVATION.PY · MEASUREMENTS.JSON

Same unit definition

```
groups=np.where(ctrl,cons,genes)[keep]
StratifiedGroupKFold, 5 folds, seed 17
shared_target_units per fold: 0, 0, 0, 0, 0
```

The key also holds out target genes and control constructs. Solver and key agree on the population (19,455 cells) and the design.

DURING · KEY_DERIVATION.PY (RESOLVED) · ANSWER_KEY.JSON

One fixed number from one tree ensemble

```
ExtraTreesClassifier(n_estimators=200,
max_features=13,
min_samples_leaf=2, class_weight='balanced')
"value": 0.7263, "absolute_tolerance": 0.025
```

The solver's first-pass mean, **0.7218**, is 0.0045 from the key and would have passed. The brief asks for “an adequately fitted predictor” and names no model class.

AFTER · KEY VARIANT RESULTS (GRADER/REFERENCE)

Where 0.7651 sits against the key builder's checks

```
13 leakage-free tree variants    0.7034 - 0.7421
cell-split (leaky) variants      0.7640, 0.7778
solver, nested SVM, leak-free    0.7651
```

The tolerance separates the leaky cell-split results from the tree variants. The solver's leakage-free SVM falls on the leaky side. No variant in the key used a non-tree model.

WHAT HAPPENED

GPT-6 Astra handled eligibility, held-out units, weighting and sequencing-group correction as specified. Its first estimate (0.7218) was within the tolerance. After tuning regularization it reported 0.7651, outside 0.7263 ± 0.025 , and the three criteria that depend on C-002 failed with it.

TRAINING TARGET

Model: when a held-out estimate moves by more than its own stated uncertainty under tuning (here 0.689–0.771 across C on one split), report that it depends on the classifier and give the spread next to the point value. **Task:** score this field against a range that covers leakage-free model choices, or score the evaluation design, before using the run as a training signal.