FORMAT & OUTPUT GUIDE

RaRaMa

Send a structure. Read the dossier.

How to format your compounds or proteins, and how to read every number that comes back. Deterministic and structure-only, from Hans-Made Research Inc.

Hans-Made Research Inc. · Winnipeg, Canada · hansley.ca

Client Data Format & Output Guide

Hans-Made Research Inc. – how to send us your compounds or proteins, and how to read what comes back.

You send a list of **structures** (small molecules) or **protein sequences**. The platform returns, from structure alone – no recipe, no panel, no wet-lab run, no prior data on your molecule:

- **Molecules** → **taste**, **aroma/smell**, **food-grade safety & stability** – each with the **mechanism behind the call** and a **confidence**.
- **Proteins** → the full **analytical-instrument dossier** – denaturation Td, DSC/TGA, CD, fluorescence, zeta, SEC-MALS/DLS, NMR, LC-MS, gelation & rheology, solubility, DIAAS – **each figure rendered in its real instrument's native format**, a same-day screen to set against your bench.

Deterministic and structure-only: the same input always returns the same numbers. The fold gate routes each sequence to its recognised fold family and returns its Td **as a confidence-tiered band** – **HIGH** on the calibrated cupin core (blind MAE ~2.7 °C, a tight ±4 °C), **PROVISIONAL** for other recognised folds (a wider, **disclosed ±band** – e.g. a heme globin ±10 °C, heme-state dependent – so a lower-confidence fold is never dressed up as a false-precision point). The value stays deterministic; the ± is the fold-family's characterised uncertainty. The gate **refuses** (emits no Td) only when a sequence is genuinely outside every recognised fold – a disordered or un-placeable chain – rather than guessing. (This is the applicability-domain discipline used by regulatory-grade QSAR and by AlphaFold's per-residue confidence: emit in-domain predictions with their band, refuse the truly out-of-domain.)

Quickstart – one job, end to end

The whole contract in four steps. The only thing you *must* give us per row is a **structure**.

1. **Write your rows.** A `smiles` (molecule) or a `sequence` (protein) each; add your own `id` to join the results back to your sheet.
2. **Pick a tier** in `options`: `1` = taste · `2` = + aroma & shelf-life · `3` = + food-grade safety & stability.
3. **Send the JSON.** We echo each structure back as an InChIKey for you to confirm – nothing runs until you do.
4. **Get back**, deterministically: a branded **PDF dossier**, a `*_results.csv` (one row per compound, your `id` is the join key), and a **SHA-256 integrity seal** you can verify yourself.

The smallest valid submission:

```
{
  "client": "Acme Foods – bitter screen",
  "compounds": [ { "id": "C001", "smiles": "Cn1cnc2c1c(=O)n(C)c(=O)n2C", "name": "caffeine" } ],
  "options": { "tier": 1 }
}
```

→ returns `caffeine → Bitter (confidence 0.93) · off-note yes · mechanism "xanthine bitter" · shelf-life years`. Re-run it tomorrow and the numbers are identical – the same input always returns the same output.

> New to the format? Skip to the **worked-example gallery** below for one filled example of every job type, copy the closest one, and swap in your structures.

---

1 · What you send

The only thing required per compound is a structure. Give it any one of these:

field	what it is	example
`smiles`	a SMILES string (most common)	`O=Cc1ccc(O)c(OC)c1`
`inchi`	a **full** InChI string – we decode it to structure	`InChI=1S/C8H10N4O2/c1-10-4-...`
`inchikey`	an InChIKey (resolved against our local map; send the full InChI if unsure)	`RYYVLZVUVIJVGH-UHFFFAOYSA-N`
`code` / `sequence`	an amino-acid letter code – we fold it to 3D and read it	`FFVAPFPEVFGK`
`cas`	a CAS registry number – resolved to structure (offline map → PubChem)	`121-33-5`
`fema`	a FEMA flavour number – resolved to structure	`2289`

> **A `smiles` or full `inchi` is the reliable input** – the system converts it to structure deterministically, **offline, with no database**, so a novel compound works exactly like a known one. `cas` / `fema` / `name` / `inchikey` are convenience **lookups** (offline map, then best-effort online) that **can miss a novel or uncommon compound** – so for anything unusual, include the `smiles` or `inchi`. Names alone are never a reliable structure source. A row with no resolvable structure is flagged in QC, never silently dropped.

Everything else is **optional**: `id` (echoed back so you can join results to your own sheet – if you omit it we echo the SMILES), `name` (display only), `measured` (your own measured label – the engine predicts from structure alone and never reads it, so it scores itself honestly against it; this drives the **validation confusion matrix** – taste vs your measured value), `role` + `fraction` (for masking / blend jobs), `pick` / `spotlight` (**mark the special compounds you want to track** – give a label like `"pick": "hero sweetener"` or just `"yes"`; every batch chart then **rings and labels those compounds** so they pop out of the 500+ cloud, and they get a **Spotlighted-compounds** table up front showing all three channels – taste · aroma · safety – in one view; this also drives the Tier-3 deep-dive isolation), and any of your own columns (passed back untouched).

JSON (the contract)

```
{
  "compounds": [
    { "id": "C001", "smiles": "O=Cc1ccc(O)c(OC)c1", "name": "vanillin" },
```

```
    { "id": "C002", "inchi": "InChI=1S/C8H10N4O2/c1-10-4-9-6-5(10)7(13)12(3)8(14)11(6)2/h4H,1-3H3" },
    { "id": "C003", "code": "DE", "name": "Asp-Glu peptide", "measured": "umami" }
  ],
  "options": { "tier": 3 }
}
```

- `options.tier` – **1** = taste · **2** = + aroma & shelf-life · **3** = + food-grade safety & stability.

VOC / aroma screens – your abundances are half the read

For an aroma / off-note screen, give each volatile its structure and add **`concentration_ppb`** – your measured GC-MS abundance (ppb = µg/kg or µg/L). From structure alone each VOC returns its **odour character** and its **detection threshold** (`otv_ppb`); your abundances are the other half. Supply them and the engine computes **OAV = your concentration ÷ that threshold**, maps it to **perceived intensity** with a structure-derived dose curve, and ranks the batch so `fd_potency_rank = 1` is the molecule actually carrying the smell. Your abundance data is a **core input, not an afterthought** – it drives the driver ranking. We read your **identified peaks and interpret the GC-MS table; we don't replace the MS**. A row with no `concentration_ppb` still returns its character + threshold.

field	what it is
`concentration_ppb`	your measured / relative abundance of that volatile (ppb = µg/kg or µg/L) – turns on `oav`, `odor_active`, the perceived-intensity read and the batch driver ranking

```
{ "compounds": [
  { "id": "V001", "name": "hexanal", "smiles": "CCCCCC=O", "concentration_ppb": 5000 },
  { "id": "V002", "name": "(E)-2-nonenal", "smiles": "CCCCCC/C=C/C=O", "concentration_ppb": 40 }
], "options": { "tier": 2 } }
```

Matrix salt (optional). Add a top-level `salt_species` (e.g. `Na2SO4`) to model ion-specific aroma release: a kosmotrope salts a hydrophobic off-note out of solution into the headspace, **raising its effective OAV at the same abundance** (see the *Ion-specific (Hofmeister)* row above). The `…_Na2SO4` worked example in the package shows it.

Masking, blends & dose-aware jobs

For a **masking** job, mark the off-note active and list the candidate maskers (group them with a shared `set_id`):

field	what it is
`role`	`active` (the off-note to fix) or `masker` (a candidate to test)
`off_note`	`yes` on the active – the bitter / astringent compound to suppress
`masker_type`	for a masker: `sweetener` / `blocker` = receptor suppression · `encapsulant` / `coating` = physical sequestration (e.g. cyclodextrin)
`use_level`	your dose – a number + unit (`0.5 mg/mL`, `15–60 ppm`, `2 pct`); drives the dose-aware intensity and the masker dose-titration curve
`matrix`	the food base it sits in (`water` / `dairy` / `pea protein` / `beverage`)

`fraction`	for a blend: each component's % share

```
{ "compounds": [
  { "set_id": "M1", "role": "active", "name": "quinine", "smiles": "COc1ccc2nccc(C(O)C3CC4CCN3CC4C=C)c2c1",
  { "set_id": "M1", "role": "masker", "name": "NaCl", "smiles": "[Na+].[Cl-]", "masker_type": "sweetener"
  { "set_id": "M1", "role": "masker", "name": "b-cyclodextrin", "smiles": "OCC1OC2...", "masker_type": "enc
] }
```

We return the **masker shortlist** – each candidate ranked by predicted % off-note reduction with the **named mechanism** (receptor suppression vs physical sequestration) – and a **4-parameter dose-titration curve** (the achievable ceiling + the practical dose). Two same-taste components (e.g. MSG + a 5′-nucleotide, or two sweeteners) are read for **synergy** instead of suppression.

CSV (the bulk interchange)

One row per compound; the **only required column is `smiles`** (or `inchi` / `code`). Add `id`, `name`, `measured`, and your own columns freely.

```
id,name,smiles,measured
C001,Vanillin,O=Cc1ccc(O)c(OC)c1,sweet
C002,Genistein,O=c1c(-c2ccc(O)cc2)coc2cc(O)cc(O)c12,bitter
```

> InChI strings contain commas – if you use a CSV, **quote the InChI cell**, or just use JSON.

Lot sizes – 100 / 500 / 5,000 / 10,000

Same file, same submission. Small lots come back in full; **large lots (5k–10k) return a preview on screen and the complete set as a downloadable file** – there is no small per-submission cap. Throughput is a fixed cost per compound (deterministic, no queue): ~50 ms/compound, **10,000 in one job**, zero rows dropped.

> **You don't need the compound's name.** Everything is computed from structure. The name is only a label on your report – leave it blank and we still return the full read.

1B · Proteins – the analytical-instrument lane

Send a protein by **sequence** or **UniProt accession** and the platform returns the full biophysical / techno-functional dossier – the read a multi-instrument characterisation bench produces, computed from structure alone.

Minimum protein row: one of `sequence` / `accession`, plus the fold and the disulfide count.

field (+ synonyms)	what it is	example
`sequence` · `seq` · `aa_sequence`	the 1-letter amino-acid sequence	`"MRAR…"`

`accession` · `uniprot` · `acc`	UniProt ID → sequence resolved & cached (send the sequence itself for a novel/undeposited protein)	`"P04776"`
`structure` · `oligomer`	fold / assembly state	`"11S hexamer"` · `"7S trimer"` · `"12S"`
`n_disulfide` · `disulfide_per_protomer`	S–S bonds per protomer (0 for 7S, 1 for 11S)	`1`
`feedstock` · `crop`	source material	`"pea"`

Device-parameter inputs – set them the way you set your bench

Each virtual instrument accepts the **same control a researcher dials on the real device**, so the prediction lands in *your* units and conditions. Most are optional with sensible defaults – but a few **materially change accuracy**, so send them whenever you know them (flagged **ACCURACY** below). A novel protein still computes with none.

> **The three that most affect accuracy** – send these when you can: the **assembly state** (`structure`/`oligomer` + `n_disulfide`, which set the cooperativity and can move Td by ~10 °C), the **cofactor/metal state** for a metalloprotein (`heme_state`, up to ~25 °C), and – always – the **mature chain** (submit the processed sequence, not a signal-peptide/pro-form precursor, or the mass, ε280 and fold call drift). Omit them and you get a labelled default, not a wrong-but-unflagged number.

Real instrument	Input field(s)	What the control drives	
DSC (TA / Mettler)	`scan_rate_C_min` · `heating_rate` · `scan_rate` (°C/min)	the Kissinger scan-rate shift – the apparent Td at *your* ramp (a faster ramp reads a higher apparent Td)	
Metalloprotein cofactor – **ACCURACY** (globin/heme · parvalbumin·α-lactalbumin/Ca · transferrin·lactoferrin/Fe)	`cofactor_state` · `heme_state` · `metal_state` · `ca_state` · `fe_state` = `"holo"` \	`"apo"` · `iron_saturation` (0–1)	the **metal-clamp melting point**: a metal-bound (holo) protein is **15–25 °C more heat-stable** than the metal-free (apo) form. **Defaults to holo – set it, or a metalloprotein Td can be off by ~25 °C** (α-lactalbumin apo 35 → holo 64; ovotransferrin apo 62 → holo 81). Any of the state synonyms works; also drives the nanoDSF cofactor-saturation readout (S1/S2 lobes).

Assembly state – ACCURACY	`structure/oligomer` (`"11S hexamer"`/`"7S trimer"`/`"dimer"`…) · `n_disulfide`	the assembly cooperativity bump – a hexamer can read up to ~10 °C above a monomer; also sets SEC-MALS Mw and the disulfide-aware mass / ε280 / pI.
Process / extruder	`target_temp` · `die_temp` (°C) · `moisture_pct` · `sme_kj_kg` · `screw_rpm` · `residence_time` (min)	survive / denature / texturize verdict + the reactive (RMD/REX) regime
Solution / buffer	`pH` · `ionic_mM` · `salt_species`	Td (moisture / Flory-Huggins depression), solubility minimum at pI, zeta Debye length; naming the salt species unlocks the ion-specific (Hofmeister) readout below
Ion-specific (Hofmeister) – *ion identity, not just strength*	`salt_species` (e.g. `Na2SO4` · `NaCl` · `NaSCN` · `sodium citrate`) + `ionic_mM`	Ranks your salt on the **Hofmeister series** (kosmotrope ↔ chaotrope) and applies its ion-specific direction across four readouts at once: **protein Td** (kosmotrope salts-out/stabilises, chaotrope salts-in/destabilises), the **isoionic→measured-IEF** shift (a chaotropic anion adsorbs and drags the electrophoretic pI below the structure-only isoionic point), **fatty-acid / interfacial adsorption**, and **volatile salting-out** (a kosmotrope raises a hydrophobic off-note's headspace → higher effective OAV at the same dose). Two salts at the *same* `ionic_mM` now read differently – SO_4^{2-} ≠ Cl^- ≠ SCN^-. Opt-in: omit `salt_species` and you get the prior ionic-strength-only behaviour, unchanged.
UV-PDA (Waters ACQUITY)	`conc_mg_ml` · `pathlength_mm` · `stray_light_pct`	A280 quantitation + Beer-Lambert linearity / dynamic range
LC-MS / IMS (Waters)	`drift_gas` (N_2 / He) · `cone_voltage` · `collision_energy_ev`	CCS (gas-specific) + in-source fragmentation
Viscometer / rheometer	`conc_mg_ml` · `temp_c`	intrinsic viscosity [η] + processing viscosity vs concentration

Gel rheology	`measured_gel_pa` @ `measured_gel_conc_pct`	one-point calibration that pins the absolute G′ scale to *your* rheometer (the rank order holds regardless of the absolute scale)
Water-activity / osmometry (AquaLab)	`use_level` (a **bulk** dose – `%`, `mg/mL`, `mM`; trace ppm/ppb is ignored)	the single-solute water activity a_w + osmolality + the microbial-safety verdict at that dose (food-safety / shelf-life)
Density-sound-velocity (Anton Paar DSA)	`conc_mg_ml` · `temp_c`	speed of sound c, adiabatic compressibility β_s = 1/(ρc²), partial specific volume ν̄ – the hydration read; defaults to 50 mg/mL / 25 °C
Thermal transport / LFA (food matrix)	`composition` (mass-% `water`/`protein`/`fat`/`carb`) *or* `moisture_pct`	thermal conductivity k, diffusivity α, cp (Choi–Okos) with the freezing discontinuity – the heat-processing read
Physicochemical panel (small molecules)	`pH` (for logD / pKa speciation)	boiling point, vapor pressure, melting point, logP/logD, aqueous solubility, pKa, molar refractivity – off distillation / DSC / shake-flask / titration / refractometer
Retrodiction (INTERNAL validation only)	`lit_td_c`	your measured/published Td → an **internal-only** predicted-vs-measured parity check for *our* validation runs; **never shown in any client or showcase deliverable** (a client has no reference to correlate against). The showcase edition instead shows our **established** reference-set validation.

Group your run controls in a `conditions` block. The pH / ionic strength / DSC scan rate / cofactor state / moisture that describe *how you ran the sample* belong together in a `"conditions": { … }` object – that block is the "mirror your bench" control set, and supplying it is what lets the prediction correlate with *your* measurement rather than a generic default. (For backward compatibility these same keys are still read if you place them at the top level of the row, but the grouped form is clearer and is what we recommend.) Identity fields (`accession`/`sequence`, `structure`, `n_disulfide`) and process fields (`target_temp`, `sme_kj_kg`…) stay at the top level.

```
{
  "client": "Your Company – pea/faba isolate screen for high-moisture extrusion",
  "compounds": [
    { "id": "ISO-1", "name": "Pea legumin 11S", "accession": "P15838",
      "structure": "11S hexamer", "n_disulfide": 1, "feedstock": "pea",
      "conditions": { "pH": 7.0, "ionic_mM": 200, "scan_rate_C_min": 10, "moisture_pct": 30 },
      "process": "extrusion", "target_temp": 150, "sme_kj_kg": 650, "residence_time": 45 },
```

```
    { "id": "LEGH-1", "name": "Soy leghemoglobin (metalloprotein)", "accession": "C6ZK17",
      "structure": "monomer",
      "conditions": { "pH": 7.0, "cofactor_state": "holo" } }
  ],
  "options": { "tier": 3 }
}
```

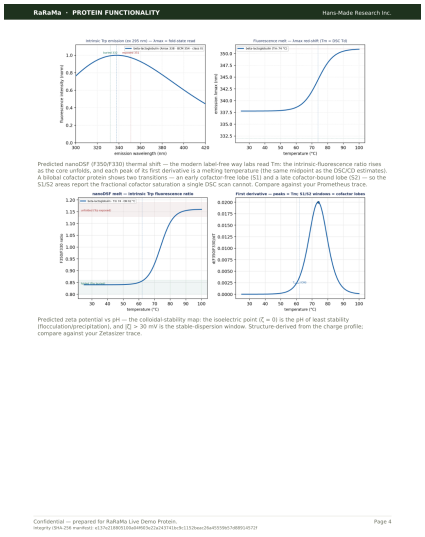
> **Numbers must be numbers.** Send `"moisture_pct": 30`, not `"moisture_pct": "30%"` – a value the engine expects numeric, sent as text, is treated as *unset* (you get a labelled default, not an error). Units are fixed: pH unitless, `ionic_mM` in mM, `scan_rate_C_min` in °C/min, `moisture_pct` as a percent number, temperatures in °C.

What the protein dossier returns – the instrument cascade, per tier

One structure → the whole bench, each figure in its instrument's **native** convention (CD in MRE units, zeta as ζ-vs-pH with the isoelectric point at the zero-crossing, SEC-MALS with molar mass across the elution peak, LC-MS as the $[M+nH]^{n+}$ charge envelope, and so on):

Tier	Protein instruments returned
T1 · baseline	fold applicability + **Td** (denaturation, reported as its disclosed **±band** with a **HIGH / PROVISIONAL** confidence tier) + **pI** – with an explicit **refusal** (no Td) when the fold is genuinely out of scope
T2 · product-readiness	+ **DSC** thermogram & extraction/gelation window · **CD** (secondary structure + θ222 melt) · **intrinsic fluorescence** (tertiary) · **zeta** (ζ vs pH, IEP) · **SEC-MALS / DLS** (Rg, Rh, Mw, PDI) · **gelation** (LGC, G′) · **solubility** vs pH
T3 · full dossier	+ **TGA / DMA** · **rheometer** (G′ sweep, gel-set) · **viscometer** ([η]) · **NMR** (DOSY diffusion, ^{15}N T1/T2) · **SAXS / Kratky** · **LC-MS** (intact mass, charge envelope, CCS, tryptic map) · **FTIR amide-I + Raman** (secondary structure, fold-gate-consistent with CD) · **density–sound-velocity** (β_s / hydration) · **DIAAS** nutrition · emulsification / foaming / WHC / OHC · multivariate PCA / dendrogram / radar

Every protein figure is a **structure-derived estimate to compare against your instrument trace – a same-day screen, not a substitute** for the bench.



An actual protein-functionality dossier page (β-lactoglobulin) – intrinsic Trp fluorescence (the fold-state read), the zeta-

vs-pH colloidal-stability map with the isoelectric point marked, and the DOSY self-diffusion / ^{15}N NMR tumbling read – all computed from the sequence.

---

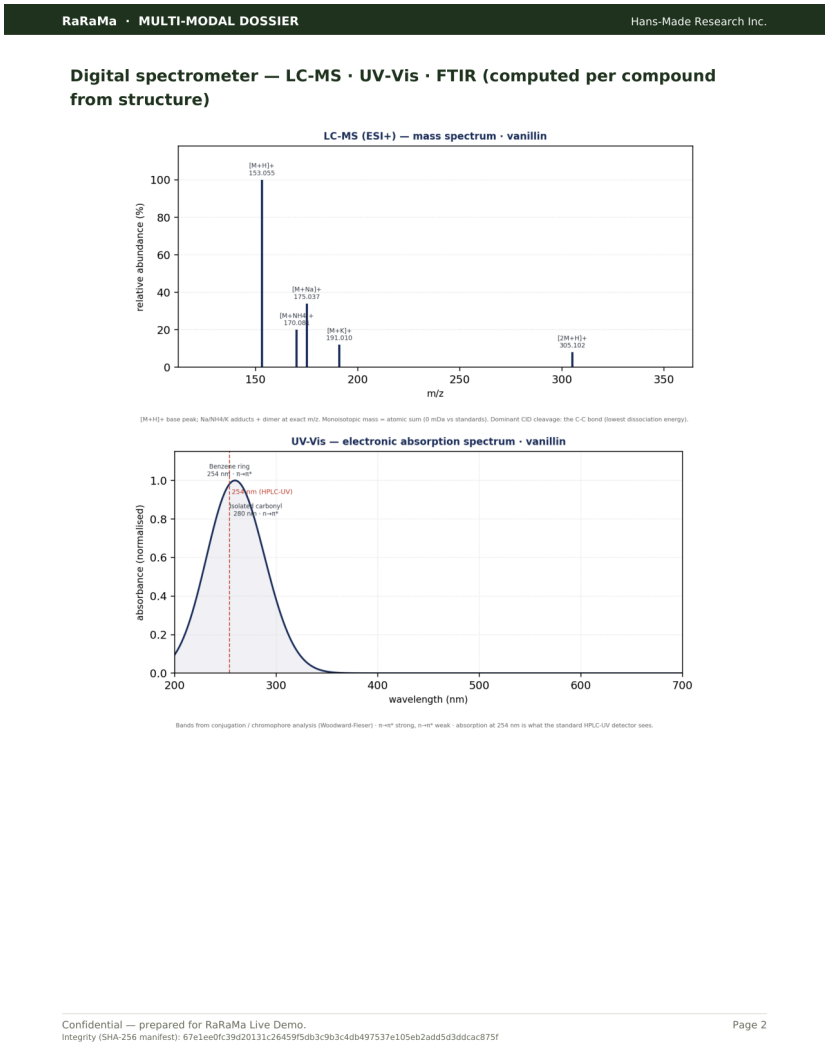
2 · What you get back

A · `*_results.csv` – one row per compound (your `id` is the join key)

column	meaning
`id`, `name`	your identifiers, echoed back
`input`	how we read it – `SMILES` / `InChI` / `sequence` / `InChIKey`
`canonical_smiles`	the structure we evaluated
`taste`	the primary taste call (sweet / bitter / umami / sour / salty / cooling / pungent / tasteless)
`secondary`	a secondary taste, if present
`off_note`	`yes` = a bitter / astringent liability flagged
`confidence`	0–1 reliability of the call
`mechanism`	the **structural reason** the call fired (the auditable "why")
`aroma`	the smell read (tier 2+) – odorant notes or "odorless / non-volatile"
`shelf_oxidative`	oxidative shelf-life class (tier 2+)
`food_safety`	food-grade safety verdict (tier 3) – PASS / CAUTION / FLAG
`mw`	molecular weight
`sweetness_equiv`	**potency vs table sugar** – sweetness relative to sucrose=1 (e.g. sucralose ≈ 600×). The formulation cheat-code: how little you need
`bitterness_equiv`	bitterness relative to quinine=1 (the bitter reference)
`otv_ppb`	**odour detection threshold** in ppb – from structure (published value where one exists, else a structural read-across screen). The denominator of the OAV
`oav`	**odour activity value** = your `concentration_ppb` ÷ `otv_ppb`. > 1 = above threshold. Blank if you didn't supply an abundance
`intensity_dose`	the **perceived odour intensity** at your supplied abundance – a structure-derived dose→intensity read, not a linear scaling of concentration
`odor_active`	`yes`/`no` – is it an active odorant (OAV > 1, the Rule-of-1)? Blank if you didn't supply a use concentration
`fd_potency_rank`	**aroma-driver rank** (1 = the strongest aroma driver in your batch, by OAV) – hands the flavourist the 3–5 molecules that actually carry the smell
`ti_onset`	Time-Intensity onset speed – `fast` / `medium` / `slow`
`ti_persistence`	aftertaste persistence – `Quick fade` / `Moderate` / `Lingering`

`dual_tail`	`yes` – a delayed secondary taste emerges as the first fades (e.g. sweet → lingering bitter – the Stevia liability)
`ti_t_max_s`	time to peak intensity, **seconds**
`ti_duration_s`	total perceived duration, **seconds**
`ti_aftertaste_s`	aftertaste duration after the peak, **seconds**
`ti_crossover_s`	for a dual-tail: when the secondary overtakes the primary, **seconds**
`masking_recommendation`	the suggested intervention to silence a lingering off-note (e.g. bulk-sweetener shielding)
`qc_flag`	`ok`, or `DROPPED_NO_STRUCTURE` / `ENGINE_ERROR` – **nothing is ever silently dropped**
`measured`	your own measured label, if you supplied one – scored against, never used as a prediction input

B · A branded **PDF dossier** (the part the field mostly lacks)



An actual dossier page (vanillin) – the LC-MS mass spectrum, the UV-Vis absorption spectrum, and the FTIR functional-

group fingerprint, each computed from structure, with the SHA-256 integrity seal in the footer.

Per-compound read + the mechanism behind each call; **positive flavour leads** (the sweet/umami/spicy candidates, not just the off-notes); your **scored margins** (sensitivity / specificity / MCC + the false-positive / false-negative count) if you sent `measured` labels – scored live on **your** compounds, not a stored claim. The dossier also renders the full sensory metrics, computed from structure alone:

- **Sensory fingerprint (QDA spider)** – the multi-attribute taste/aroma profile, scored 0–10 per descriptor, as the page-1 hero for a deep-dive compound.
- **Kinetic liabilities (Time-Intensity)** – the predicted dynamic curve in **seconds**: how fast each compound arrives (onset), where it peaks (T_max), and how long it lingers. Dual-tail compounds (e.g. a sweetener that decays into a rising bitter aftertaste) are plotted as two channels with the crossover marked, and carry a **masking recommendation**.
- **Formulation & aroma drivers** – potency equivalents (×-sweeter-than-sucrose / ×-quinine-bitterness) and the odour-active FD-ranked drivers (ranked by **OAV = your abundance ÷ the structural threshold**) – the few molecules to actually dose and tune.

C · For large batches – the chemical-space map

Batches of 500+ also get a **library map** (UMAP **and** PCA projections), coloured by predicted taste and **sized by confidence** so the high-confidence non-bitter leads stand out, with **cluster specs** and the **compound–compound interaction** flags that drive a real recipe.

D · QC, never silent

Any row with no readable structure, or one that errors, is **flagged in the CSV** (`qc_flag`) and listed in a QC ledger – never quietly dropped (a known failing of some services).

E · The complete delivery package – one self-contained folder

Every run returns a **self-contained, independently-verifiable folder** (`clients/<your-project>/`). Nothing is scattered; everything needed to read, reuse, and audit the run travels together:

file	what it is
the call log (`submission_<job>.txt`)	the human-readable **computation log** – per compound, the taste/off-note call + the *why*, the aroma, shelf-life, and every instrument readout (LC-MS, UV-PDA, FTIR, NMR, physchem) in one plain-text pass. This is the record of exactly what the engine computed.
the PDF dossier (`RaRaMa_report_<job>.pdf`)	the branded, client-safe report with every figure (§B).
the results JSON (`submission_<job>.json`)	the machine-readable **JSON of the results** – every field behind the PDF, for your own pipeline.
the results CSV (`*_results.csv`)	the flat table, one row per compound (§A).
your input (`<your-file>.json`)	the exact panel you sent, bundled back so the package reproduces itself.

the integrity seal (`*_merkle.json`)	the SHA-256 manifest – re-run and it reproduces; change one atom and it changes.
`_assets/`	every figure as a PNG (the aroma wheel, QDA spider, Time-Intensity curves, instrument traces, library map) – each **computed live from your compounds**, never a stock image.
`_delivery_index.json`	the manifest: what was delivered, the tier, the compound count, the seal hashes, and the one-line command to verify it yourself.

(Protein jobs also include per-protein `computation_ledgers_<job>.json` – the full deterministic derivation of every instrument value, step by step.) Deterministic and offline: push the same JSON again and the whole package regenerates identically, seal and all.

Worked examples – one filled example of every job type

Copy the block closest to your job, swap in your structures, keep the shape. Each is a **real submission** the platform runs; the `→` line is the call it actually produced.

A · Taste / off-note screen (tier 1)

```
{ "client": "…", "options": { "tier": 1 }, "compounds": [
  { "id": "T1", "smiles": "Cn1cnc2c1c(=O)n(C)c(=O)n2C", "name": "caffeine", "measured": "bitter" },
  { "id": "T2", "smiles": "OC[C@H]1O[C@@](CO)(O[C@H]2O[C@H](CO)[C@@H](O)[C@H](O)[C@H]2O)[C@@H](O)[C@@H]1O",
] }
```

→ `caffeine → Bitter 0.93 (off-note)`, `sucrose → Sweet 0.75`. Supplying `measured` makes the run **score itself** against your label (never read as an input) – this drives the sensitivity / specificity / MCC panel.

B · Aroma + shelf-life (tier 2)

```
{ "options": { "tier": 2 }, "compounds": [
  { "id": "A1", "smiles": "COc1c(Cl)cc(Cl)cc1Cl", "name": "TCA (cork taint)", "measured": "musty" }
] }
```

→ the taste call **plus** the aroma character (musty/moldy · aroma-active) and the oxidative shelf-life class.

C · Bitter masking (a set – group with a shared `set_id`)

```
{ "options": { "tier": 1 }, "compounds": [
  { "set_id": "M", "role": "active", "name": "caffeine", "smiles": "Cn1cnc2c1c(=O)n(C)c(=O)n2C", "off_note
  { "set_id": "M", "role": "masker", "name": "NaCl", "smiles": "[Na+].[Cl-]", "masker_type": "sweetener" },
  { "set_id": "M", "role": "masker", "name": "homoeriodictyol", "smiles": "COc1cc(C2CC(=O)c3c(O)cc(O)cc3O2)
] }
```

→ a **ranked masker shortlist**, each with a predicted % off-note reduction, the named

mechanism (receptor suppression vs physical sequestration), and a 4-parameter dose-titration curve.

D · Protein dossier (tier 3)

```
{ "options": { "tier": 3 }, "compounds": [
  { "id": "P1", "name": "Pea legumin 11S", "accession": "P15838",
    "structure": "11S hexamer", "n_disulfide": 1, "feedstock": "pea",
    "conditions": { "pH": 7.0, "ionic_mM": 200, "scan_rate_C_min": 10, "moisture_pct": 30 } }
] }
```

→ the full instrument cascade – Td (± band, HIGH/PROVISIONAL), DSC, CD, fluorescence, zeta/pI, SEC-MALS/SAXS, viscosity, NMR, LC-MS, FTIR – each in its instrument's native units.

Do this – not that

✗ Not that	✓ Do this
`"smiles": "C1CC1("` (unbalanced ring)	a valid, parseable SMILES – a malformed row is flagged in QC, never guessed
`"moisture_pct": "30%"` (text)	`"moisture_pct": 30` – a number sent as text is read as *unset* → a labelled default
a name only, no structure	include a `smiles` / `inchi` / `sequence`; a name alone can't resolve a novel compound
a signal-peptide / pro-form protein	the **mature** processed chain – or the mass, ε280 and fold call drift
a metalloprotein with no cofactor state	set `cofactor_state` / `heme_state` – it moves Td up to ~25 °C (defaults to holo)

Field vocabularies, validation & QC

Controlled vocabularies (the values the engine accepts / returns)

Field	Allowed values
`options.tier`	`1` (taste) · `2` (+ aroma & shelf-life) · `3` (+ safety & stability)
`role` (within a `set_id`)	`active` · `masker` · `component` (blend) · `control` (QDA benchmark)
`masker_type`	`sweetener` / `blocker` (receptor suppression) · `encapsulant` / `coating` (physical sequestration)
`taste` (returned)	`sweet` · `bitter` · `umami` · `sour` · `salty` · `cooling` · `pungent` · `tasteless`
`off_note`	`yes` (a bitter / astringent liability) – else blank
`ti_onset` / `ti_persistence`	`fast`/`medium`/`slow` · `Quick fade`/`Moderate`/`Lingering`
`food_safety` (tier 3)	`PASS` · `CAUTION` · `FLAG`
`structure` / `oligomer`	`monomer` · `dimer` · `7S trimer` · `11S hexamer` · `12S` …

`cofactor_state` / `heme_state`	`holo` (default) · `apo`
`salt_species` (ion-specific)	a named salt/ion: `Na2SO4` · `(NH4)2SO4` · `sodium citrate` · `NaF` · `NaCl` · `KCl` · `NaBr` · `NaNO3` · `NaI` · `NaClO4` · `NaSCN` · `GdnCl` … (kosmotrope → chaotrope). Omit for ionic-strength-only

Validation rules (checked before anything runs)

- **Structure required.** Each row needs one of `smiles` / `inchi` / `inchikey` / `cas` / `fema` / `name` (molecule) or `sequence` / `accession` (protein). A `smiles`/`inchi` resolves offline for a novel compound; `cas`/`fema`/`name`/`inchikey` are convenience lookups that can miss an uncommon one.
- **Numbers must be numbers.** `"moisture_pct": 30`, not `"30%"` – a numeric field sent as text is treated as *unset*, never an error.
- **Units are fixed.** pH unitless · `ionic_mM` in mM · `scan_rate_C_min` in °C/min · `moisture_pct` a percent number · temperatures in °C.
- **Mature chain for proteins**; group blends/masking sets with a shared `set_id`.
- **Nothing is silently dropped** – every *N* rows in are accounted for as *N* rows out.

QC / error catalog (the `qc_flag` column)

`qc_flag`	Meaning	Fix
`ok`	scored cleanly (the InChI / sequence decode is shown in the report)	–
`DROPPED_NO_STRUCTURE`	no readable structure in the row	add a `smiles` / `inchi` / `sequence` and re-submit that row
`ENGINE_ERROR`	reached the engine but failed to compute	check the structure is chemically valid – the row is listed in the QC ledger, not discarded

Every dropped or errored row is also written to a QC ledger, so you always get a full accounting – the "descriptive, not punitive; nothing lost" convention.

---

3 · How to read the result

- **Three channels, one submission**: taste, smell, and safety together – no service stitches these for you.
- **A named mechanism on every call** – not a black-box score; you can see *why* it's bitter, *which* group fires.
- **A confidence on every call**, and **no applicability-domain cliff**: being structure-only and deterministic, we return a real call on novel / peptide / food chemistry where trained models report "out of domain".
- **Deterministic** – the same structure returns the same answer every run, no model drift (auditable QA).

Accuracy – the numbers, on public benchmarks

On the public ChemTastesDB (N ≈ 3,500 tastants, held out, zero training): **six-class taste accuracy 70.7%**, the honest "predict the taste class" number, against a **33.6% shuffled-label null**, with **off-note detection about 82% balanced accuracy** and the **umami call at MCC 0.68** on held-out peptides (iUmami UMP442, n=88). Alongside a third-party University of Manitoba e-tongue **contracted correlation** of R^2 0.95 (above its own 0.94 calibration).

Because we never train on labels, a high score cannot be data leakage – it reproduces from structure. We publish the **shuffle-control** that collapses these numbers to chance; that collapse is the evidence the result is derivation, not memorisation.

This is a predictive screen and decision-support layer – not a replacement for your own evaluation.

---

4 · FAQ

Do you need the molecule's name? No. Everything is from structure; the name is just a label.

What about our proprietary structures? They travel only inside an encrypted (HTTPS/TLS) channel, behind a per-client API key; the engine runs on our side and is never shipped to you. You receive the results, not the method.

What if a compound is an edge case? Predictions carry a confidence, and any scored run reports its false-positive / false-negative margins, so you can see exactly where the calls are strong vs uncertain. **Whether – and where – to confirm anything physically is entirely your decision; we provide the screen and the confidence, not the verdict.** We are decision support, not a replacement for your own process.

Can you tell us how to fix a bitter off-note? Yes – we flag which off-notes are "reprofilable" and can guide a formulation that silences the bitterness while keeping the core (the underlying method is proprietary).

Versioning, integrity & the portal

Deterministic & version-pinned. The same input, on the same engine version, always returns the same dossier – no model drift. Every deliverable carries an **engine SHA-256, an input/results hash, and a timestamp** in its footer.

Verify it yourself. Each run ships a Merkle integrity sidecar (`*_merkle.json`): recompute the root from the per-compound leaves, or re-submit the same file and confirm the manifest hash matches. Any change in transit – an input or a computed result – changes the seal.

Applicability domain – we refuse rather than guess. In-domain, a value ships **with its ± band** (e.g. `Td = 84 ± 6 °C`), never a bare false-precision number. A genuinely out-of-domain input – an unplaceable fold, an unreadable structure – is **refused with the reason**, not

dressed up as a confident point. Refusing the wrong question is part of keeping the right answers trustworthy.

Decision support, not a replacement. Every call names its mechanism so you can independently review the basis. Use it to prioritise, triage and design experiments; confirm the ones that matter at the bench.

The portal (coming soon)

A self-serve portal is in early access. Until it opens we onboard every lab personally – send your JSON/CSV by email and we return your first dossier plus a short walkthrough. **Everything you format now maps 1:1 onto the portal schema, so nothing is throwaway.** When it opens you'll get:

- a **blank template** + a **filled example** to start from,
- **schema validation on upload** – every issue at once, each with the field, what we expected, and how to fix it,
- a **test mode** – submit a sample structure, get a fully-rendered dossier watermarked DRAFT, no charge, so you learn the format risk-free before you go live.