Die-Link Analysis — Handoff Briefing

Roman Republican didrachms RRC 26/1 and RRC 27/1. Bipartite die-link graphs, crossing minimisation, and multi-anvil workshop hypotheses.

Purpose of this note: everything needed to resume the work in a fresh conversation without re-deriving anything. Paste this file in, along with the artifact links below.

---

1. Sources

Type	CRRO record	Die study
RRC 26/1	`https://numismatics.org/crro/id/rrc-26.1`	attributed to R. Schaefer
RRC 27/1	`http://numismatics.org/crro/id/rrc-27.1`	attributed to R. Schaefer

Both are anonymous silver didrachms from the mint of Rome. RRC 26/1 is dated 234–231 BCE (laureate head of Apollo right / horse galloping left, ROMA). RRC 27/1 is dated c. 230–226 BCE. The die-link table sits under the **Die Analysis** heading on each page, as a three-column list: obverse die, reverse die, number of recorded specimens.

Data provenance caveat. The RRC 26/1 table in §3 was transcribed directly from a live fetch of the CRRO page and cross-checked against the die URIs (`rrc-26.1.o.1`–`o.32`, `rrc-26.1.r.1`–`r.36`), which confirm 32 obverse and 36 reverse dies. The RRC 27/1 table in §4 was transcribed earlier in the same working session and has **not** been re-verified against the source since; re-fetch and diff it before relying on any 27/1 figure.

---

2. Artifacts already built

All are self-contained HTML pages saved to the account. To continue work on one, paste its link into a new chat or Cowork session and ask for changes; each edit saves a new version.

RRC 27/1

1. Clustered die-link graph (initial ordering) — `https://claude.ai/code/artifact/a4990ef0-a09d-41e1-8ca5-77f7b9659b05`
2. Same graph, crossing-minimised ordering — `https://claude.ai/code/artifact/f4a5ddd6-77f3-4cb8-8320-3a81ed340b63`
3. Two-anvil hypothesis, chain A — `https://claude.ai/code/artifact/a67458a4-274f-41f3-95dc-c92b295e11fc`
4. Three-anvil hypothesis, chain A — `https://claude.ai/code/artifact/40053b59-7fef-479a-a1da-9a7ab66087c1`

5. Three-anvil hypothesis, chain B — `https://claude.ai/code/artifact/9cd72dd6-6cf7-43d0-aa70-77d88b2f306d`
6. Two-anvil hypothesis, chain B — `https://claude.ai/code/artifact/a98190a0-2c30-47ed-8ca6-cd38d3cf42ea`

RRC 26/1

7. Clustered die-link graph, crossing-minimised — `https://claude.ai/code/artifact/99cbb8f9-a59a-48d9-85f3-bd58c6018e46`
8. Two-anvil hypothesis, all three affected chains — `https://claude.ai/code/artifact/4da4ada7-0219-4cbd-a2c3-d0f7c735273d`

---

3. RRC 26/1 data

Totals: 56 pairings · 32 obverse dies · 36 reverse dies · 116 recorded specimens · 13 connected die-chains.

Naming trap. Obverse and reverse dies both use numeric names in the 1000s *and* they are unrelated. Obverse die 1000 is not reverse die 1000. Obverse dies are numbered 1–14 and 1000–1017; reverse dies are lettered A–Z, AA–AJ, plus 1000–1004. Always carry an `O`/`R` prefix internally.

Pairings (obverse, reverse, count)

```
1-A 13      1-AA 1      10-1004 1   10-C 1      1000-D 3
1001-1003 1 1001-D 1    1001-X 1    1002-C 1    1003-AH 1
1004-I 2    1005-H 1    1005-I 1    1006-Q 2    1007-N 2
1008-G 2    1008-M 1    1009-AG 1   1009-C 1    1010-U 1
1011-X 2    1012-A 3    1012-AA 4   1013-AC 2   1014-B 1
1015-I 1    1016-D 2    1017-1002 1 11-C 1      11-K 5
12-K 3      13-U 2      13-Y 3      14-1000 4   14-P 1
2-AD 2      2-B 3       2-L 4       2-Q 1       2-V 2
3-B 1       4-1001 1    4-AJ 2      4-T 4       5-AE 2
5-J 3       6-B 2       7-S 3       7-T 1       8-1001 2
8-E 1       8-H 3       8-L 1       8-W 2       9-AF 2
9-Z 1
```

Connected components (die-chains)

Chain	Obverse dies	Reverse dies	Pairings	Coins
A	2, 3, 4, 6, 7, 8, 1004, 1005, 1006, 1014, 1015	B, E, H, I, L, Q, S, T, V, W, AD, AJ, 1001	23	43
B	10, 11, 12, 1002, 1009	C, K, AG, 1004	8	14
C	1000, 1001, 1011, 1016	D, X, 1003	6	10
D	1, 1012	A, AA	4	21

Chain	Obverse dies	Reverse dies	Pairings	Coins
E	13, 1010	U, Y	3	6
F	14	1000, P	2	5
G	5	AE, J	2	5
H	1008	G, M	2	3
I	9	AF, Z	2	3
J	1013	AC	1	2
K	1007	N	1	2
L	1003	AH	1	1
M	1017	1002	1	1

---

4. RRC 27/1 data — re-verify before use

Totals as used: 43 pairings · 20 obverse dies · 21 reverse dies · 6 chains.

```
1002-D 1   1002-N 1   1002-T 2   1004-D 1   1008-N 1   1010-D 2
3-D 1      3-E 1      3-H 1      3-T 1      4-P 3
5-N 1      5-P 4      5-S 1
6-E 3      6-H 2      6-L 1      6-T 1
7-D 1      7-E 1      7-H 1      7-T 2
10-H 1     10-L 1
1-A 1      1-B 1
9-1000 1   9-B 3      9-C 1      9-M 1
2-C 3      2-F 3      2-J 1      2-V 2      2-W 1
1001-F 2   1001-V 2   1000-M 2
1003-R 1   1005-O 1   1006-K 1   1007-Q 1   1009-Q 2
```

Chain	Obverse	Reverse	Pairings	Coins
A	3, 4, 5, 6, 7, 10, 1002, 1004, 1008, 1010	D, E, H, L, N, P, S, T	24	24
B	1, 2, 9, 1000, 1001	A, B, C, F, J, M, V, W, 1000	14	14
C	1003	R	1	1
D	1005	O	1	1
E	1006	K	1	1
F	1007, 1009	Q	2	3

---

5. Method

5.1 Chains

Treat the pairings as an undirected bipartite graph and take connected components. A chain is a set of dies tied together end-to-end through shared coins. Chains never share a reverse die with each other, so **every chain is an independent problem** — analyse each separately and never let one chain's layout constrain another's.

5.2 Counting crossings

In a two-layer drawing, assign each obverse die an index `a` by its position in its column and each reverse die an index `b` by its position in the other column. Two edges cross iff their indices invert:

```
def crossings(edges, oidx, ridx):
    L = [(oidx[a], ridx[b]) for a, b in edges]
    return sum(1 for i in range(len(L)) for j in range(i+1, len(L))
               if (L[i][0]-L[j][0]) * (L[i][1]-L[j][1]) < 0)
```

Ties never count — edges sharing an endpoint cannot cross. Every artifact computes this live from its own data rather than trusting a hardcoded number; keep that property when editing.

5.3 Minimising crossings

Barycentre/median heuristic (Sugiyama-style): reorder one column by the mean position of each node's neighbours in the other, alternate sides, iterate to a fixed point. Then verify with pairwise-swap hill climbing from many random restarts. For small chains, brute-force all permutations.

5.4 When zero is impossible — the useful theory

A bipartite graph has a crossing-free two-layer drawing **iff it is a caterpillar forest**: every component is a tree, and deleting all leaves leaves a simple path.

Two things therefore force crossings, and both are structural facts about the dies rather than drawing failures:

- **Cycles.** RRC 26/1 chain D is a complete 2×2 (obverse 1 and 1012 each struck against both reverse A and AA). One crossing is forced.
- **Branch points.** A node whose links fan into three or more non-trivial sub-chains cannot be flattened. The cost of a branch point is roughly the edge count of the smallest sub-branch you are forced to span — so when ordering, always span the *cheapest* branch.

Known branch points: RRC 26/1 obverse die 8 and reverse die C; RRC 27/1 obverse dies 3, 6, 7 (a mutually-overlapping triple) and obverse die 2.

5.5 Testing an n-anvil hypothesis

Premise: *n* obverse anvils worked simultaneously against one shared pool of reverse punches. Layout: reverse pool in a single shared column, obverse dies split into *n* groups around it.

The test is whether a partition exists in which **no anvil's own lines cross each other**, given a **single reverse ordering shared by all anvils**. That shared ordering is the whole difficulty — each side is easy alone.

Search used (`anvil2.py`, reproduced below in outline):

1. Enumerate every two-way partition of a chain's obverse dies (halve by symmetry).
2. For each, hill-climb the shared reverse ordering, re-optimising each side's obverse order at every step.
3. Record the best; prefer the most **balanced** partition among those reaching zero, since a split that isolates one die is trivially clean and archaeologically uninteresting.

```
def evaluate(rev_order, obvL, obvR, edges):
    ridx = {n: i for i, n in enumerate(rev_order)}
    eL = [e for e in edges if e[0] in obvL]
    eR = [e for e in edges if e[0] in obvR]
    return best_obv_order(obvL, eL, ridx) + best_obv_order(obvR, eR, ridx)
```

---

6. Findings so far

RRC 26/1

- Single-column optimised layout: **7 crossings**, all structural. Chain A 5, chain B 1, chain D 1; the other ten chains draw cleanly.
- Chain A's 5 was confirmed near-optimal by 300 randomised restarts with full pairwise-swap hill climbing — nothing ever beat it. Chain B's 1 was brute-forced as the true minimum (2,880 orderings). Chain D's 1 is mathematically forced.
- **Two anvils clears all seven.** Most balanced zero-crossing splits found:
    - Chain A, 5/6 — anvil 1: 1004, 1015, 8, 4, 7 · anvil 2: 1005, 3, 6, 1014, 2, 1006 · reverse order: I, H, E, W, B, L, 1001, AD, AJ, V, T, S, Q
    - Chain B, 2/3 — anvil 1: 1002, 10 · anvil 2: 1009, 11, 12 · reverse order: AG, C, 1004, K
    - Chain D, 1/1 — anvil 1: 1 · anvil 2: 1012 · reverse order: A, AA

RRC 27/1

- Chain A **cannot** be resolved with two anvils: 33 crossings unsplit, 10 remaining after the best two-way split, because obverse dies 3, 6 and 7 mutually overlap (3 and 7 share all four of their reverse dies; 6 shares three with each).
- Chain A **does** resolve fully at three anvils — 3, 6 and 7 each take their own — giving zero crossings across all 24 pairings.
- Chain B resolves fully at **two** anvils (anvil 1: 1, 9, 2 · anvil 2: 1000, 1001); the only constraint is separating die 2 from die 1001, the sole pair sharing more than one reverse punch. A third anvil does no work there.

The interesting comparison

RRC 26/1 accommodates a two-anvil workshop effortlessly. RRC 27/1 chain A structurally refuses to and needs a third. If the two types were struck in the same workshop within a few years of each other, that difference wants an explanation.

---

7. Standing caveats — keep stating these

- A crossing-free arrangement shows the pairing record is **consistent** with *n* simultaneous workstations. It does **not** establish that *n* existed. A single anvil cycling through dies over time produces an identical record. Planarity is a necessary condition for the hypothesis, never a sufficient one.
- Any given clean split is one of several valid ones. The specific dies assigned to "anvil 1" are a demonstration of the method, not a claim about which dies shared a workbench.
- Prefer balanced splits. A partition that isolates a single high-degree die is trivially clean and says little.
- Schaefer's die attributions are themselves interpretive, and survival is heavily biased by what reached auction catalogues. Coin counts are counts of *recorded specimens*, not of coins struck.
- Report what was actually verified versus what was heuristically found. Say "brute-forced minimum" only where a full enumeration was run.

---

8. Artifact conventions

Worth preserving for visual consistency if you extend the set:

- Self-contained HTML, no external assets. Colours defined as CSS custom properties on `:root` with `prefers-color-scheme: dark` overrides plus explicit `[data-theme]` blocks, so pages render correctly in either theme.
- Serif type stack (Iowan Old Style / Palatino / Georgia), warm paper background `#f6f2ea`, ink `#2a2420`.
- Node radius scales as `4.6 + sqrt(coin_count) * 2.3`; edge stroke width scales with specimen count for that pairing.
- Chain palette: `--c1` blue `#3d6098`, `--c2` orange `#b8632f`, `--c3` green `#4a7a55`, `--c4` purple `#6a5391`, `--c5` red `#a33f3f`, `--c6` muted `#8a8072` for small chains.
- Every `<circle>` and `<path>` carries a `<title>` for hover detail. Crossing counts are recomputed in-page.
- Anvil diagrams: reverse pool centred, obverse dies flanking; unresolvable overlaps drawn as dashed red lines rather than hidden.

---

9. Obvious next steps

- Re-fetch RRC 27/1 and diff against §4.
- Extend to neighbouring types (RRC 28/1 onward) and ask whether chains ever bridge across types — shared reverse punches between types would be a much stronger workshop signal than anything internal to one type.
- Weight the analysis by specimen count rather than treating every pairing equally; a 13-coin link and a 1-coin link currently carry the same structural weight.
- Test whether the 27/1 chain A triple (3, 6, 7) is better explained by sequential re-use than by simultaneity — a chronological ordering of dies by wear or weight might discriminate.
- Consider a die-wear or weight-based sequence if the Schaefer plates support it, which would let the anvil hypotheses be tested against something other than topology alone.