

Validation Report — Restriction Site Detection

Work: codex-vivum · **Version:** v1.0 · **Date:** 2026-05-12 **Author:** Thomas Schüller · ORCID 0009-0003-9799-6747 · CC BY 4.0

This report implements section 4.2 of the validation pre-registered in the paper. It also documents a defect found and fixed in the process.

Design

The tool's site detection was ported to Python and compared against `Bio.Restriction` from Biopython — an independent implementation whose enzyme data derive from REBASE.

Test set. 19 sequences (two of them real cloning vectors): twelve random sequences across lengths of 1, 3, 5 and 10 kb at GC contents of 35, 50 and 65 per cent, plus five constructed edge cases (site at the start, at the end, across the seam of circular DNA, closely spaced sites, ambiguous patterns). Deterministic via fixed random seed (4273).

Scope. 43 enzymes, all present in `Bio.Restriction`. One comparison per enzyme and sequence for linear and one for circular DNA: 1,462 position comparisons, plus 166 comparisons of fragment sizes.

Finding — overlapping sites were missed

The first run yielded 99.18 per cent agreement. All deviations concerned a single enzyme: **NlaIV**, recognition sequence GGNNCC. The tool consistently found *fewer* sites than the reference.

Checking the 28 additional sites showed they are valid GGNNCC matches. Example from a test sequence: at GGGTCCC two matches overlap — GGGTCC from position 76 and GGTCCT from position 77. The tool's regular expression continued past the end of a match and skipped the second.

This was a genuine defect. A real cut site was not shown, and the fragments computed from it were wrong accordingly.

Extent. A systematic test of all 43 enzymes showed that seven patterns can overlap themselves: ClaI (ATCGAT), NheI (GCTAGC), NotI (GCGGCCGC), NsiI (ATGCAT), PvuI (CGATCG), SphI (GCATGC) and NlaIV (GGNNCC). For the six unambiguous ones, overlap is so rare in random sequences that it did not occur in the test set — the defect was latent there and would have surfaced on real sequences.

Fix. The search pattern is now wrapped in a lookahead, which consumes nothing; the scan then advances by one position rather than by the match length. Overlapping sites are found as a result.

Two further findings

After the overlap defect was fixed, continuing the check revealed two further problems, both concerning the informativeness of the check itself.

The enzyme list was kept twice

The first version of this script contained a hand-copied second version of the enzyme table. Comparison with the source revealed nine discrepancies: enzymes were checked that do not exist in the tool (NlaIV, ApoI, BstYI, NsiI, SacII, SspI, MspI), while nine enzymes actually present (HpaII, MboI, MfeI, PacI, PciI, SbfI, BsaI and others) were never covered. The previously reported figure therefore referred in part to a library that does not exist in that form.

Fixed by having the script read the table from `src/lib/restriction.js`. If the file differs or is missing, the run aborts. A second copy of the same data is no longer kept.

Type IIS enzymes cut incorrectly on the bottom strand

The run over the actual library yielded 99.04 per cent. All deviations concerned BsaI, not in the number of sites but in their position: every bottom-strand site was off by exactly 4 nucleotides.

BsaI is a type IIS enzyme and cuts outside its recognition sequence, per REBASE GGTCTC(1/5) — one nucleotide beyond the site on the top strand, five on the bottom. The tool's data model knew only *one* cut offset. For a recognition site on the bottom strand the second is required. The difference of 4 is exactly the distance between the two offsets.

Fixed by an additional `cutBottom` field in the enzyme entry and separate arithmetic for type IIS enzymes on the bottom strand. A test ensures no future type IIS enzyme is entered without this second value.

Assessment of these two findings

The first concerned the check, not the tool — but it came close to reporting a wrong result as confirmation. The second concerned the tool and would never have surfaced without fixing the first, because BsaI was absent from the hand-copied list.

Result after the fix

Check	Comparisons	Agreement
Cut positions, linear DNA	817	817 (100.00 %)
Cut positions, circular DNA	817	817 (100.00 %)
Ambiguous recognition sequences	3 enzymes	3 (100 %)
Fragment sizes	178	178 (100.00 %)

Fragment lengths sum to the sequence length in every case.

Comparison with published data

The cloning vectors **pUC19** (2,686 bp, GenBank L09137) and **pBR322** (4,361 bp, J01749.1) were added to the test set. Their provenance and the checks performed before use are documented in `sequenzen/HERKUNFT.md`.

This yields a comparison against a **third** source: for pUC19 it is published that the ten polylinker enzymes EcoRI, SacI, KpnI, SmaI, BamHI, XbaI, SalI, PstI, SphI and HindIII each cut exactly once. The tool finds exactly one site for all ten. This check does not depend on the reference implementation but on the published map of the

vector.

Total after adding the vectors: **1,634 position comparisons**, all identical.

What this report does not show

- The comparison was against an independent implementation whose enzyme data come from the same source (REBASE). Agreement evidences correct use of the data, not its correctness.
- Positions and fragment lengths were checked. The simulated agarose gel display and the overhang information were not part of this check.
- A comparison against NEBcutter itself is still outstanding; the checks were against Bio.Restriction and against the published polylinker map of pUC19.
- Methylation dependence (as for DpnI) is not modelled by the tool and was not part of the comparison.

Assessment

The value of this check lies less in the confirmation than in the finding. Without comparison against an independent implementation the NlaIV defect would not have surfaced — it manifests not as a crash but as the silent absence of individual sites.

Reproduction

```
pip install -r requirements.txt
python3 restriktion_validation.py
```

The script exits with status 0 only on complete agreement. Full output: `restriktion_output.txt` .