



# LiftOn 2.0: scaling accurate genome annotation to whole genomes

By Kuan-Hao Chao

Jun 17, 2026 · 10 min read

## Abstract

LiftOn 2.0 is a major update to the genome-annotation lift-over tool that combines DNA alignment (Liftoff) and protein alignment (miniprot). The core idea is unchanged: use both signals, then choose the gene model that best preserves the reference protein. The update focuses on making that idea reliable at whole-genome scale. It adds a memory-bounded windowed aligner for giant genes, a best-of-outcome merge that keeps the DNA-plus-protein model only when it improves the emitted protein, default lifting of gene-like feature types beyond protein-coding genes, full-genome robustness fixes, and faster execution protected by a 24-cell byte-identity regression matrix. Across the 18 benchmarks where both LiftOn versions ran to completion, LiftOn 2.0 matched or exceeded v1.0.8 on all 18 and matched or exceeded the best single-method baseline on 17 of 18, with human-to-mouse as the one accuracy non-win. On full Arabidopsis and rice RefSeq genomes, it finished runs the previous release abandoned and recovered tens of thousands more coding transcripts.

---

LiftOn was built around a simple observation: DNA alignment and protein alignment fail in different ways. DNA lift-over with Liftoff preserves gene structure well when two genomes are close, but it can introduce frameshifts, premature stops, and shifted exon boundaries as sequence diverges. Protein-to-genome alignment with miniprot survives divergence better, but it can be fragmentary and does not naturally preserve the reference transcript structure.

LiftOn combines the two. For each gene, it compares the Liftoff model and the miniprot model against the reference protein, chains together the coding segments that best preserve that protein, and then runs an open-reading-frame rescue step to repair boundary problems. The first release showed that this dual-evidence strategy improves protein-coding annotation transfer across genomes and across species.

**LiftOn 2.0** keeps that scientific idea, but rebuilds the tool for the way people now use genome annotation: whole RefSeq genomes, complete assemblies, and many target species at once. The update is less about adding a new headline method and more about making the method dependable at scale.

# What changed

The first release was accurate, but it was engineered around chromosome-scale runs. Whole genomes exposed four practical problems. A few giant genes could drive full dynamic-programming alignment into tens of gigabytes of memory. Some full plant genomes crashed partway through, leaving partial annotations that looked usable but were not complete. The pipeline spent too much time in serial execution and disk round-trips. And the default lift covered protein-coding genes while quietly leaving pseudogenes, long non-coding RNA genes, tRNAs, rRNAs, and other gene-like features behind.

LiftOn 2.0 addresses those problems with four main updates.

First, it replaces the most memory-hungry alignments with an **anchor-windowed aligner**. LiftOn scores candidate genes by aligning their translated proteins back to the reference protein. A full Needleman-Wunsch alignment with traceback is quadratic in memory, so one giant gene can dominate the entire run. The new aligner finds exact shared anchors, chains them into a consistent backbone, and runs full dynamic programming only inside the windows between anchors. For ordinary homologous genes, this keeps the same answer while bounding peak memory. On the largest memory cases, the practical change is dramatic: the benchmark shows controlled same-input runs using up to about 24x less memory, and the most memory-intensive four-way mouse subset drops from 44.6 GB in v1.0.8 to 0.76 GB in v2.0.

Second, it makes the DNA-plus-protein merge more conservative. The old LiftOn always emitted the merged model after protein maximization. That is usually right, but a local improvement can still perturb a downstream reading frame. LiftOn 2.0 now evaluates the merged model and the DNA-only model after ORF rescue, then emits the one with the better reference-protein identity. In other words, miniprot evidence is used when it improves the final transcript, not simply because it exists. On identical aligner inputs, that change has the shape we wanted: many improvements and almost no regressions, including 115 improved transcripts and 1 regressed transcript on the fruit-fly benchmark, and 83 improved with 0 regressed on mouse-to-rat.

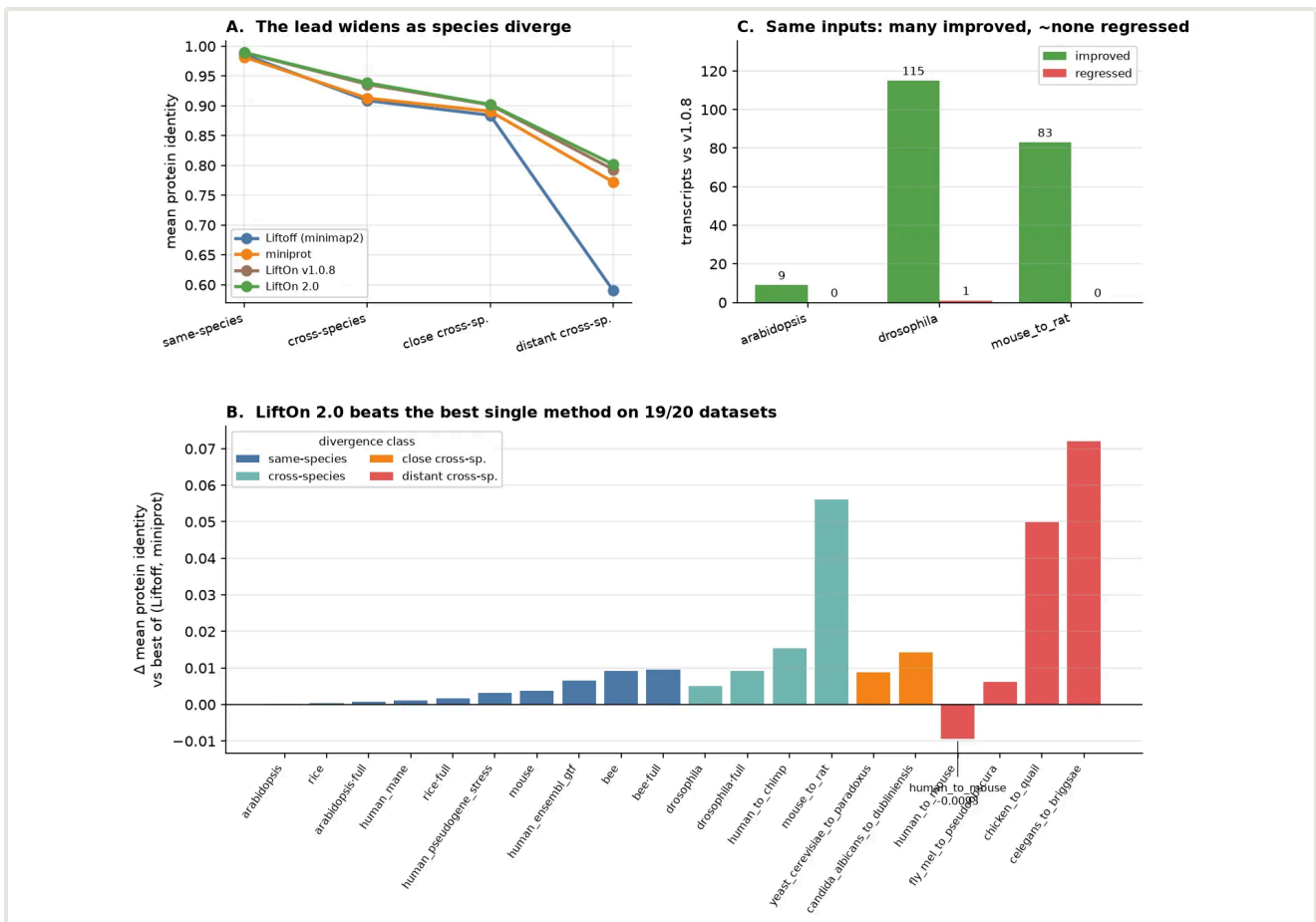
Third, LiftOn now lifts **every gene-like feature type** by default. Instead of assuming only `gene` parents matter, it scans the reference annotation for top-level feature types that behave like genes: a parent record with transcript and exon children. That brings pseudogenes, non-coding RNA genes, tRNAs, rRNAs, snoRNAs, snRNAs, and related structured features into the lift. The change matters because these features are part of the annotation, not decoration. Dropping them means downstream analyses inherit a narrower genome map than the reference actually provides.

Fourth, the execution engine was rebuilt under a byte-identity contract. The two external aligners can run concurrently. The per-locus merge can run across worker threads. Some intermediate files can be streamed or passed in memory. miniprot now receives the requested thread count. These are engineering changes, so they should not silently change the annotation. LiftOn 2.0 protects that boundary with a 24-cell regression matrix: every combination of streaming, in-memory Liftoff handoff, thread count, and native bindings must reproduce the default GFF3 byte-for-byte. If a speed path changes one byte of output, it fails.

# What the benchmark shows

I evaluated LiftOn 2.0 on a 20-dataset benchmark spanning a divergence ladder: same-species assembly-to-assembly lift-overs, moderate cross-species pairs, close cross-species pairs, and distant cross-species pairs. The comparison includes four tools: Liftoff alone, miniprot alone, LiftOn v1.0.8, and LiftOn 2.0. Every output is re-scored by the same evaluator rather than by each tool's own reporting. For coding transcripts, the evaluator translates the lifted CDS, aligns it to the reference protein, and computes mean protein identity.

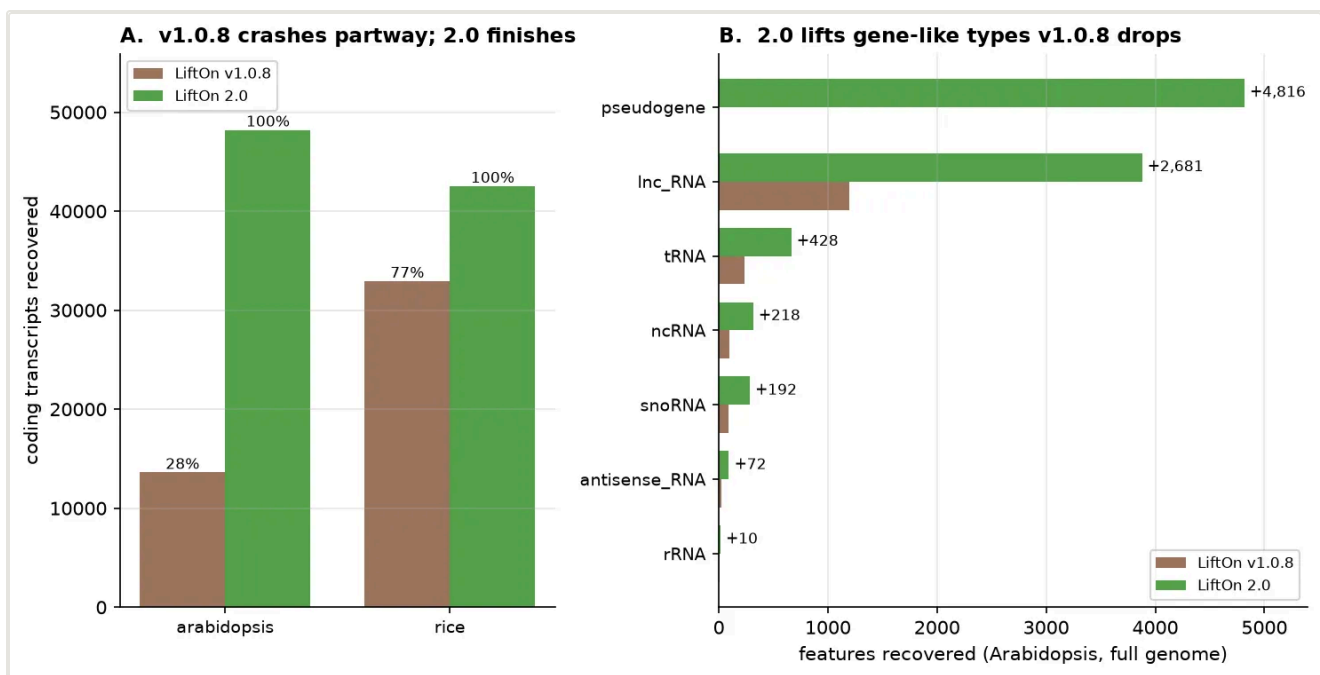
Two full genomes from v1.0.8 crashed partway through, so they are not used for the head-to-head accuracy comparison. Across the 18 datasets where both LiftOn versions completed, LiftOn 2.0 matched or exceeded v1.0.8 on all 18. It also matched or exceeded the better single-method baseline, Liftoff or miniprot, on 17 of 18. The one exception is human-to-mouse: at that distance, miniprot's protein-only model is higher by 0.00931 mean protein identity. That is an important boundary condition. Combining DNA and protein helps most when both signals still contain useful information; when the DNA signal is very degraded, protein alone can be the cleaner evidence.



**Figure 1.** Accuracy across the divergence ladder. LiftOn 2.0 is strongest where DNA-only and protein-only evidence are both imperfect but still informative. It matches or exceeds the best single-method baseline on 17 of 18 completed benchmarks; human-to-mouse is the one non-win.

The largest practical result is not an accuracy decimal. It is that LiftOn 2.0 finishes full genomes that v1.0.8 abandoned. On full RefSeq Arabidopsis, v1.0.8 crashed after recovering 13,618 coding transcripts, about 28% of the reference set. LiftOn 2.0 completed the run and recovered 48,207, adding 34,589 coding transcripts. On full rice, v1.0.8 stopped at 32,933 coding transcripts, about 77%; LiftOn 2.0 completed the run with 42,527, adding 9,594.

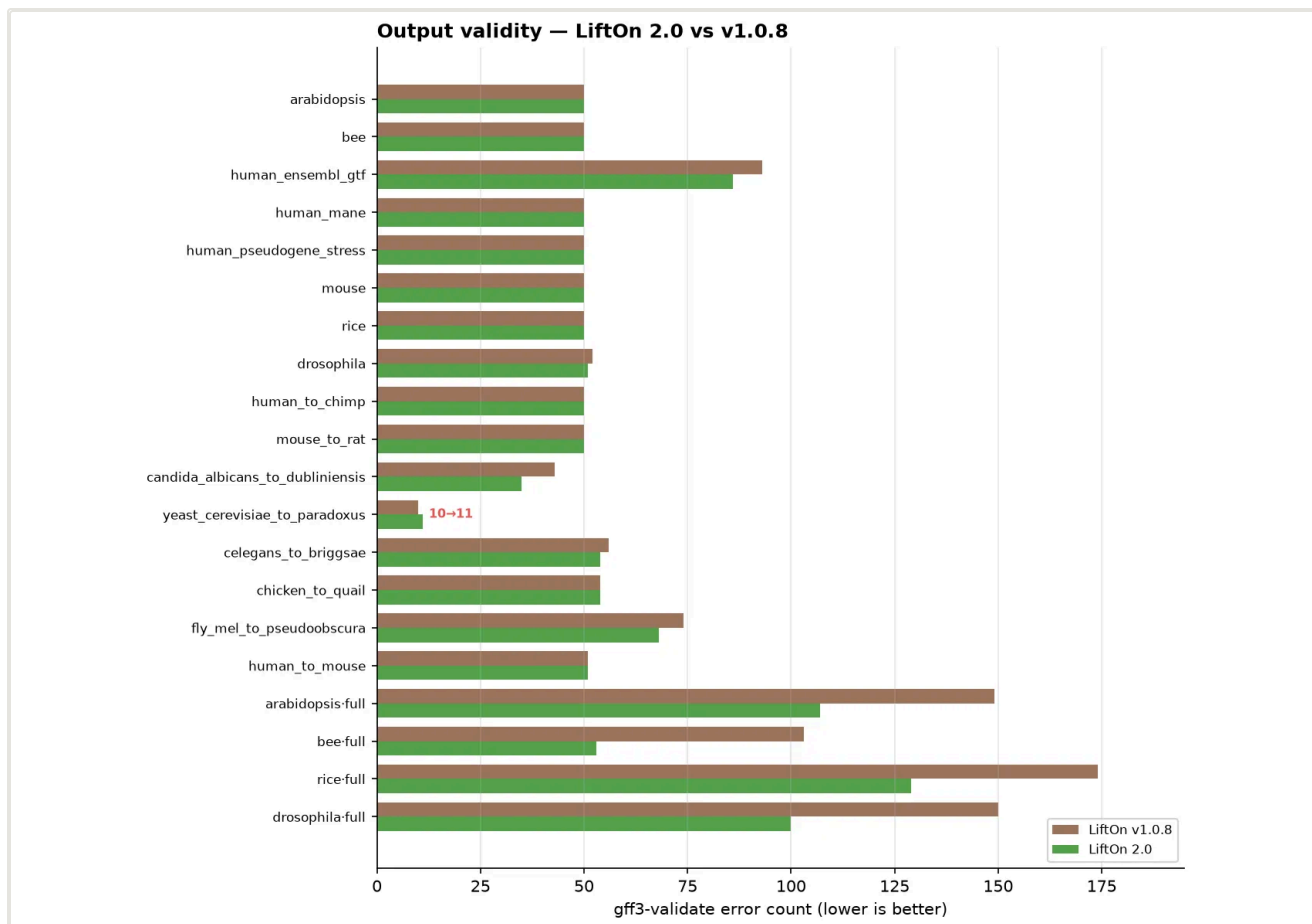
The same figure also shows the broader feature lift. On full Arabidopsis, LiftOn 2.0 recovers thousands of gene-like features that v1.0.8 omitted by design: 4,816 pseudogenes, all 3,878 Inc\_RNA genes in the reference set, many more tRNAs and snoRNAs, and nearly all rRNA and antisense RNA records. This does not mean every nested product is solved. Mature miRNA recovery remains poor in this release because those records are nested under primary transcripts in a way the current transcript-centric logic does not yet handle well. But the parent gene-like features now come through, and the annotation is much less artificially coding-only.



**Figure 2.** Whole-genome robustness and feature breadth. LiftOn 2.0 completes full Arabidopsis and rice runs that v1.0.8 abandoned, and it lifts gene-like feature classes beyond protein-coding genes.

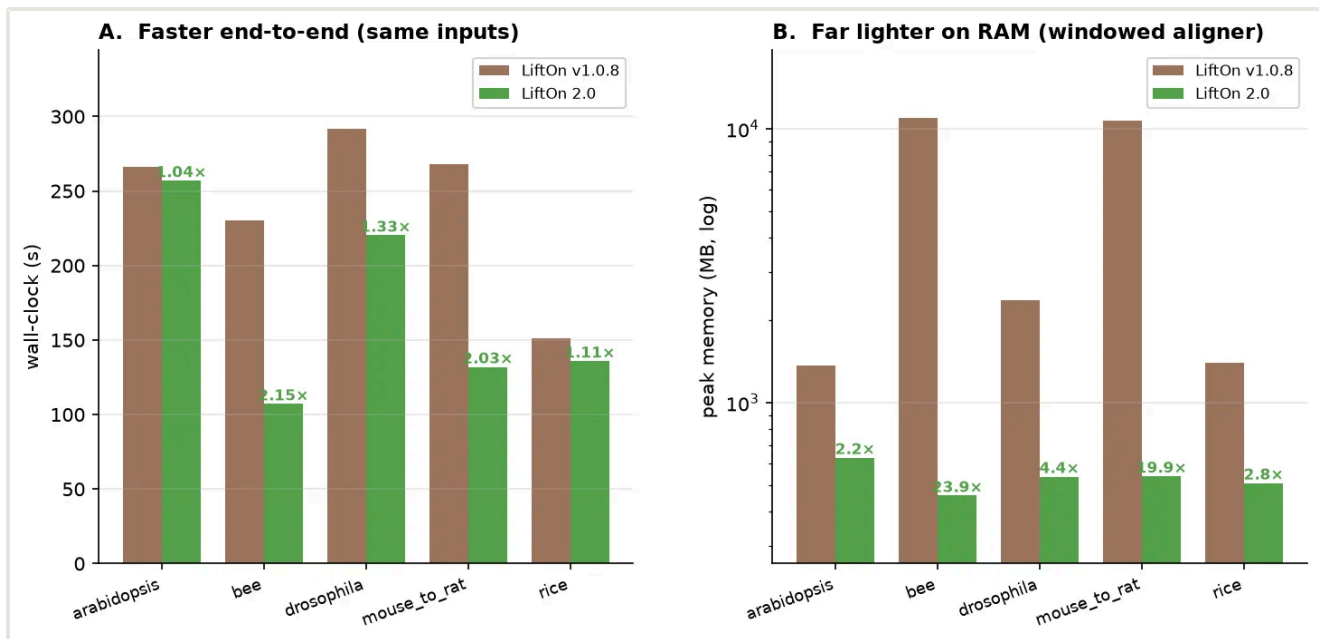
Completeness still needs a careful reading. On same-species and close cross-species data, the tools recover similar coding transcript counts. On very distant pairs, miniprot can recover more coding transcripts because it searches directly with proteins and is not anchored to a DNA lift. For example, on *C. elegans* to *C. briggsae*, miniprot recovers substantially more coding transcripts than LiftOn. LiftOn's goal is different: retain reference-consistent transcript structure, use protein evidence to rescue and correct the DNA lift, and emit accurate gene models when a reliable lift exists. On raw recall alone, protein search can win on the hardest pairs. On recovered transcript accuracy and whole-genome robustness, the combined approach is the stronger default.

Output validity also improves where scale stresses the pipeline. Re-running every output through one GFF3 validator, LiftOn 2.0 is at or below v1.0.8 on 17 of the 18 completed comparable benchmarks and is cleaner on the completed full genomes. The main caveat is small but real: on the *S. cerevisiae* to *S. paradoxus* benchmark, the validator count increases from 10 to 11 errors. That single regression is worth keeping visible because a lift-over tool is only useful if its accuracy claims and its failure modes are both explicit.



**Figure 3.** Output validity. LiftOn 2.0 generally emits cleaner GFF3 while lifting more annotation; the one observed validation regression is the yeast close-cross-species pair.

Finally, the performance numbers explain why the new version changes practical use. On identical cached aligner inputs, v2.0 is faster end to end and much lighter on memory. The largest controlled memory reductions come from the windowed aligner, while the speedups come from concurrent aligners, per-locus threading, and less disk/database overhead. On fresh whole-genome runs, v2.0 can sometimes do more total work than v1.0.8 because it lifts the expanded gene-like feature set; the important point is that the memory bound makes those runs feasible and the robustness fixes let them finish.



**Figure 4.** Faster and lighter on identical inputs. The execution changes reduce wall time, and the windowed aligner drives the large peak-memory drop.

## What changes for users

The basic command is unchanged:

```
lifton -g reference.gff3 reference.fa target.fa -o lifton.gff3
```

The defaults are more complete and more conservative. LiftOn 2.0 lifts all gene-like features by default, uses the best-of-outcome merge, and uses the memory-bounded aligner. If you need to reproduce the published LiftOn 1.x behavior, the relevant opt-out flags are `--gene-only`, `--legacy-merge`, `--full-dp-align`, and `--serial-aligners`.

For larger runs, `--threads N --locus-pipeline` enables the threaded per-locus path, while `--stream` and `--inmemory-liftoff` reduce intermediate file overhead. These speed paths are covered by the byte-identity matrix, so they are performance choices rather than analysis choices: the annotation should be byte-for-byte identical to the default output.

The main thing I want users to take from LiftOn 2.0 is that genome annotation lift-over needs both biological evidence and software discipline. The biological argument is still DNA plus protein: each signal catches errors the other misses. The engineering argument is now just as important: the tool has to finish full genomes, fit in realistic memory, lift the annotation types people expect, and make performance refactorors prove that they did not change the science.

LiftOn 2.0 is free and open source. Browse the [code](#), work through the [documentation](#), or read the [Genome Research paper](#) behind the original LiftOn method. LiftOn was built with Jakob M. Heinz, Celine Hoh, Alan Mao, Alaina Shumate, Mihaela Pertea, and Steven Salzberg at Johns Hopkins.

---

## References

1. Chao, K.-H. et al. Combining DNA and protein alignments to improve genome annotation with LiftOn. *Genome Research* (2025). [DOI](#)
2. Shumate, A. and Salzberg, S. L. Liftoff: accurate mapping of gene annotations. *Bioinformatics* (2021). [DOI](#)
3. Li, H. Protein-to-genome alignment with miniprot. *Bioinformatics* (2023). [DOI](#)