phaeno

WHITE PAPER

PSEQ WHITE PAPER SERIES · PAPER 3 OF 4

PSeq Data Pipeline

From raw FASTQ files to traceable whole-molecule RNA records

July 2026 | Version 1.0

Executive summary

The PSeq data pipeline is designed to transform paired-end short reads from a PSeq tagged RNA library into molecule-specific transcript records. The process begins by locating the PSeq marker tag, recovering the Source Molecule Identifier (SMID) and strand signal, and grouping reads that originated from one tagged reverse transcript into “molecular bins”. Within each bin, the data pipeline identifies and retrieves the sequence of the source gene, assembles sequence, creates alignment products, and records the evidence chain in a run-specific SQL database.

This paper describes the computational architecture, data products, deployment model, and molecular provenance strategy. The initial library and data pipeline traceability evidence is consolidated in Paper 4 of this series.

Computational objective: Turn a population of tagged short reads into individually inspectable molecular records that preserve all steps and intermediates back to raw sequence FASTQ files and quality evidence.

Input model

The data pipeline acquires all the raw paired-end FASTQ files directly from the sequencer together with run and sample metadata. Read 1 contains a recognizable PSeq marker tag containing wrappers, a Source Molecule Identifier (SMID), invariant check bases, and strand-sense information. The remainder of Read 1 and all or most of Read 2 contain cDNA fragment sequence. Within PSeq Clear-Signal Architecture™, the SMID converts experimentally assigned source-molecule identity into a too for organizing computation. The system first partitions read pairs into molecule-specific bins (MSBs), allowing the source gene to be identified, the gene sequence to be retrieved, and assembly, alignment, and consensus generation to operate independently within each bin individually, for the full transcript population.

Pipeline stages

1. Upload and register raw data. Receive the FASTQ files and associate them with client/sample meta data, together with the PSeq-run metadata. All uploaded and assembled information is recorded in a pipeline-generated run-specific SQL database.
2. Localize marker blocks. Locate wrapper and check-base landmarks positioned at the start of Read 1.
3. Recover molecular identity. Parse the SMID barcode and strand discriminator for each valid marker tag.
4. Cluster read pairs by SMID. Place reads carrying the same source-molecule code into an MSB.
5. Trim tag sequence. Locate and remove marker, linker and adaptor sequences, allowing that downstream searches operate exclusively on cDNA segments.
6. Deduplicate reads. Identify redundant trimmed records while retaining the evidence needed to audit the transformation.
7. Identify the source gene. Search trimmed reads and derived k-mers against an appropriate local transcript reference database and retrieve the full sequence of the leading source-gene candidate.
8. Assemble the molecule. Perform de novo contig assembly, alignments with top transcript candidates, alignments against the source gene sequence, and de novo assembly of the consensus sequence.
9. Generate analysis products. Populate files conforming to deliverables in the client run report, and which may be retrieved, interrogated and reorganized by executing SQL inquiries.
10. Create FASTA, BAM, consensus, transcript-match, and summary records.
11. Archive. Retain all information, including final reports, in the run-specific SQL database.

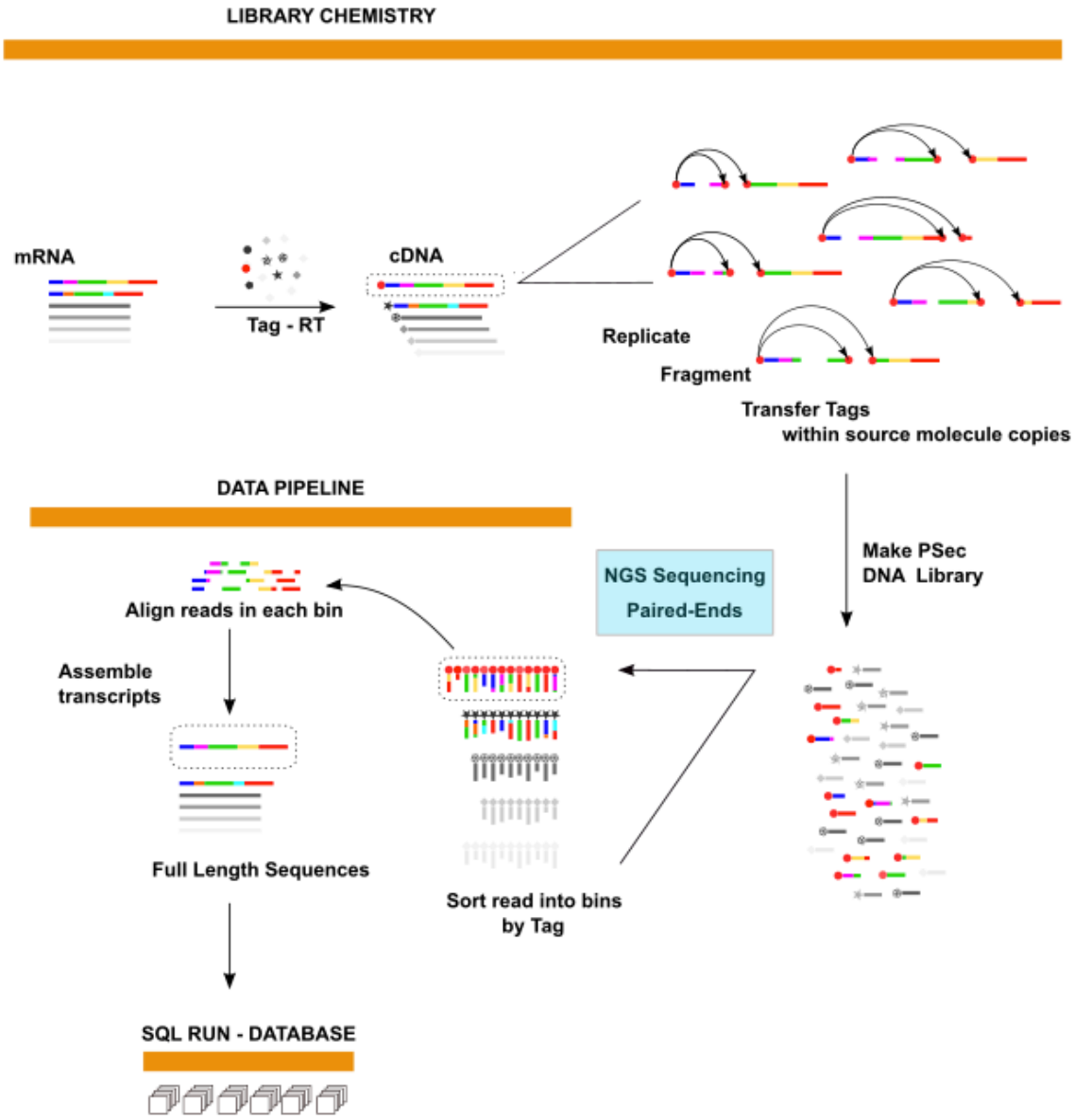

Figure 1. The data-pipeline portion of the integrated PSeq workflow groups reads by tag, assembles transcript sequences, and records outputs in a run-specific SQL database.

Gene identification and molecule assembly

Reference Search

For each MSB, the data pipeline searches trimmed reads and read-derived k-mers against a local reference database. The leading transcript or gene match provides a candidate source-gene identity and retrieves the gene sequence for downstream comparison. Appropriate reference database may include ENS curated human transcript ensembles, databases for commercial synthetic spike-in standards including as SIRV or ERCC panels, or specialty databases for alternative species or infectious agents such as RNA viruses.

De novo assembly

Preliminary assemblies are alignment maps of trimmed reads to reference transcripts (e.g., ENS transcript database), for gene identification and retrieval of source gene sequence. For this purpose, modifications of open-source Inchworm assembler (uninterrupted contigs) and open-source Trinity assembler (single consensus sequence) are fast and slow options, respectively. De novo transcript assembly is performed by trimmed read alignments against the identified source gene, to capture included and skipped exons, and discover alternative translation initiation sequence and alternative polyadenylation sites that may lie within or outside of the Ref Seq annotated gene boundary.

Paper 4 of this series confirms that assembly implements true shotgun sequencing of individual transcripts with exceptionally high consensus sequencing accuracy. Intermediate steps in data assembly are recorded in the SMID file of the run-sample SQL database created by the data pipeline.

Alignment and consensus

Trimmed reads within an MSB are aligned to the selected source-gene sequence and recorded in a BAM file. Alignment depth, exon coverage, splice-junction coverage and linkage continuity, mismatches, and base-quality evidence contribute to a molecule-specific consensus. Preliminary reference-transcript matches report length, percent coverage, and sequence agreement for comparison with annotated isoforms. Transcript alignment maps can be retrieved to profile population wide coverage, for example, as a function of transcript length.

Expected sequencing-read contract

The sample SQL database is the primary run deliverable. The database preserves the results and the path used in their production. The exact production schema remains an implementation specification, but the required information falls into several stable record layers

Record layer	Representative contents	Purpose
Client and sample	Client identifiers, sample description, project context	Connect molecular output to the submitted material
Run	Instrument/standardized run metadata, data pipeline version, configuration	Connect run specifications for run reproducibility and optimization.
Raw evidence	FASTQ file references, read identifiers, quality scores	Maintain traceability to sequencer output
Molecule identity	SMID, strand sense, bin membership	Define the source-molecule unit of analysis
Gene assignment	Top transcript matches, gene ID, retrieved reference sequence	Document source gene selection
Assembly	Inchworm/Trinity contig assembly, with alignments to reference transcripts. De novo assembly vial alignments with full-length source gene sequence.	Easily visualize blast alignments of transcripts. De novo assembly against gene sequence discovers each target transcript independently; provides guidelines when expanded gene annotation is to be registered.
Alignment	BAM output, coverage, splice-junction and mismatch evidence	Enable independent molecular visualization and quantitative review. Support automated full-sample QC validations.
Final molecular record	FASTA, consensus sequence, summary fields, external gene links	Support client reports and downstream analysis

Molecular provenance as a first-class requirement

To be confidently used for artificial intelligence (AI) training sets, data must be high quality and harbor no hidden uncertainties. Equally important, experimental data to be interrogated with AI , e.g., for diagnostic decision making or choosing individualized therapeutics, must equally conform to the highest standards of reliability.

Molecular data is most useful when the provenance of all steps in assembly is traceable and easily accessible for query. The Clear Sequence Architecture™ is optimized for traceability of molecular provenance.

This depth of information supports (1) inspection of individual molecules with open-source utilities like the IGV gene viewer, or (2) programmatic inspection to certify reliability of every record in a large data set, and generate reliable quality control summaries, detect unusual events, or generate client reports.

Deployment and scaling

The initial architecture is installed for cloud-based implementation on Amazon EC2 processors. Each environment contains the data pipeline software, local reference databases, and the run-specific SQL database. Work can be scaled by increasing the number of processors within an instance and by distributing work across multiple instances.

The v1 PSeq data pipeline operation targets a 6 to 12-hour analysis cycle, while shorter run times are likely feasible. Run-time metrics remain anecdotal until outcome distributions are reported across defined run sizes, reference databases, and ongoing compute optimizations.

Operational Requirement: Treat tag sequence specifications, library protocol, marker parsing, and the SMID/strand data model as one integrated interface. A change in any one component will trigger a compatibility and performance review.

Client reporting and downstream computation

The run-specific SQL database is designed to support client-facing reports without discarding molecule-level detail. A report can summarize gene inventories, discovered transcript and protein isoforms, coding and regulatory information bundles, new gene boundaries and alternative exons, a summarize QC indicators while preserving links to each supporting molecular record.

The structured record set may also support machine-learning and natural-language SQL query execution for specific applications. These are forward-looking use cases rather than current performance claims. Before describing PSeq data as AI-ready, data quality metrics will be formally quantified and tested.

Validation interface

Paper 4 in this series summarizes initial platform validation, confirming: (1) the structural fidelity of libraries generated with proprietary PSeq reagents and protocols; and (2) operation of the automated data pipeline, generating structures of single molecule while maintaining deep links to fundamental events between uploading raw data and discovery of the final complete structure.

A broader validation program will entail the use of synthetic reference standards, replicating data sets of hundreds of thousands to tens of millions of transcripts, with samples of varying quality, alternative instruments, and platform operators, as set out in that document.

Conclusion

The PSeq data pipeline is designed around a significant, yet simple, computational advantage created by PSeq libraries: fragments can be grouped by experimentally assigned source-molecule identity before transcript reconstruction. The pipeline then combines marker parsing, read clustering, gene identification and retrieval, assembly, alignment, consensus generation, and assembles a provenance-rich SQL data product. Its distinguishing feature is not any one algorithm, but the continuity of evidence from raw read to independently inspectable molecular record.

Notes

1. Reference-database examples in the source report include a curated ENS transcript collection and synthetic SIRV or ERCC spike-in standards; production references and versions must be controlled.
2. IGV is an external open-source alignment viewer: https://igv.org/.
3. Runtime, throughput, and data-volume statements are design targets from the internal PSeq v1 overview and require formal benchmarking.

PSeq white paper series

PAPER 1	PAPER 2	PAPER 3	PAPER 4
PSeq: Whole-Molecule Phased RNA Sequencing on Standard Short-Read NGS	PSeq Molecular Tagging and Library Architecture	**PSeq Data Pipeline**	Initial Technical Validation of the PSeq Whole-Molecule RNA Sequencing Platform.