AD Speed Profile Viewer

Algebraic Diversity for RNA Polymerase II Elongation Speed from GRO-Seq

Micah Thornton, PhD

Spring 2026

Department of Mathematics
Texas Woman's University

Outline

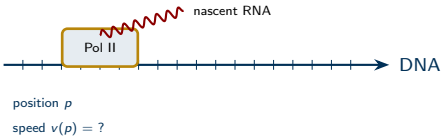
1. Biological Motivation
2. Data & Ground Truth
3. The Algebraic Diversity Framework
4. Results
5. Interactive Web Application
6. Discussion

Biological Motivation

Why elongation speed?

Transcription is not instantaneous.

- RNA Polymerase II (Pol II) walks the gene body at roughly 1–4 kb/min.
- Speed varies ∼4-fold between genes and within a gene body.
- Speed affects:
  - alternative splicing decisions,
  - co-transcriptional RNA folding,
  - pause–release dynamics,
  - response to hormonal and stress signals.
- *Measuring* speed at base-pair resolution is still hard.



Core question

Can we estimate $v(p)$ at 1 bp resolution from standard GRO-Seq data?

The occupancy-time trick

GRO-Seq captures the 5′ ends of nascent transcripts. A read at position p means a Pol II molecule was *elongating through* p at the time of run-on.

Read density is occupancy time:

$$x(p) \propto \tau(p) \quad \text{(dwell time)}.$$

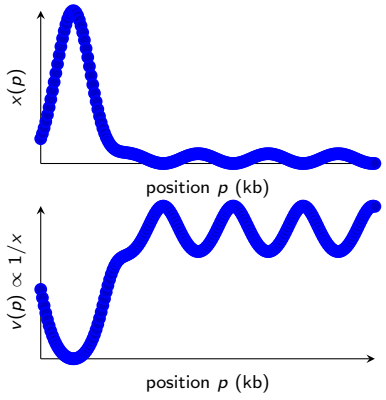
Dwell time is inverse speed:

$$\tau(p) = \frac{1}{v(p)}.$$

Therefore:

$$\boxed{v(p) \propto \frac{1}{x(p)}}$$

Reciprocal-coverage speed at every base pair.



Data & Ground Truth

Danko et al. (2013) — our benchmark dataset

Reference: Danko *et al.*, *Mol. Cell* 50(2):212–222, 2013.

Experiment

- Cell line: MCF-7 (human breast cancer).
- Treatment: 17-β-estradiol (E2).
- Time points: 0, 10, 25, 40 min, 3 biological reps each.
- Assay: GRO-Seq, single-nucleotide resolution BED records.
- Genome: hg19. GEO accession GSE41324.

Why it is perfect for this study

- Published ground-truth wave-front rates for 81 genes.
- Four-point time course $\Rightarrow$ temporal group structure.
- Replicates $\Rightarrow$ classical AD estimation averages.

Library sizes (reads)

Sample	Reads
0m R1–R3	54.4M
10m R1–R3	55.2M
25m R1, R3	30.3M
40m R1–R3	50.5M
Total	190.4M

Pre-processing

BED $\rightarrow$ RPM-normalized 1 bp coverage $\rightarrow$ group-averaged per-timepoint. Gain $\approx$ 4.77 dB for 3 reps.

Danko's wave-front method (the prior art)

Idea. E2 triggers a synchronous burst of Pol II at the promoter; its leading edge propagates downstream over time.

Procedure (`groHMM`).

1. Compute difference coverage $\Delta_t(p) = x_t(p) - x_0(p)$ for $t \in \{10, 25, 40\}$ min.
2. Fit a 3-state left-to-right HMM (upstream / wave / downstream).
3. Viterbi-decode; wave end = last position in state 1.
4. Regress wave-end position vs. time: slope = rate (kb/min).

Rate distribution (Danko):

- Median $\approx$ 2 kb/min.
- Range 1–4 kb/min.
- 4-fold gene-to-gene variation.

Limitation

Gives *one scalar rate per gene*. No information about *where* within the gene the polymerase is fast or slow.

The Algebraic Diversity Framework

Algebraic Diversity (AD) in one slide

Let $x \in \mathbb{R}^M$ be a single realization of a zero-mean random vector with covariance R. Standard estimator:

$$\hat{R}_{\text{naive}} = xx^\top \qquad \text{(rank 1, useless)}.$$

AD construction. Let G act on $\mathbb{R}^M$ by orthogonal transformations $\{P_g\}_{g \in G}$ that commute with R. Average the rank-1 estimator over the orbit:

$$\boxed{\hat{R}_G \;=\; \frac{1}{|G|} \sum_{g \in G} (P_g\, x)(P_g\, x)^\top}$$

Properties.

- Unbiased: $\mathbb{E}[\hat{R}_G] = R$.
- Full rank whenever G is large enough.
- Processing gain $= 10 \log_{10} |G|$ dB over the naive rank-1 estimator.
- Free of additional measurements — one sample, many pseudo-samples.

AD applied to a single gene: the cyclic group $\mathbb{Z}_M$

Treat the 1 bp coverage of a gene of length M as one vector $x = [x_0, x_1, \ldots, x_{M-1}]^\top$. Let P be the cyclic shift matrix.

$$\hat{R}_{\mathbb{Z}_M} \;=\; \frac{1}{M} \sum_{k=0}^{M-1} (P^k x)(P^k x)^\top.$$

Key fact. $\hat{R}_{\mathbb{Z}_M}$ is diagonalized by the DFT, and its eigenvalues are the periodogram ordinates:

$$\lambda_f \;=\; \frac{|X[f]|^2}{M}, \qquad f = 0, \ldots, M-1.$$

Processing gain. For $M = 100{,}000$ bp (typical gene body),

$$10 \log_{10}(10^5) \;=\; \textbf{50 dB}.$$

One coverage vector, full-rank spectral estimate, 50 dB over the naive estimator.

Spectral concentration ψ

Definition

$$\psi \;=\; \frac{\lambda_{\max} - \lambda_{\min}}{\mathrm{tr}(\hat{R}_{\mathbb{Z}_M})} \;=\; \frac{\max_f |X[f]|^2}{\sum_f |X[f]|^2} \quad (\text{if } \lambda_{\min} \approx 0).$$

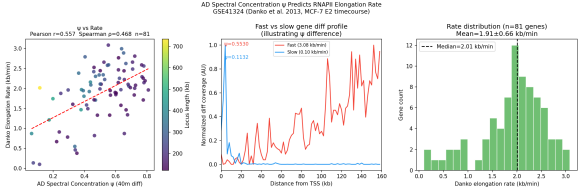
Interpretation.

- High ψ: energy concentrated in one frequency bin $\Rightarrow$ the E2-induced signal is a *broad coherent wave* filling the gene body $\Rightarrow$ fast gene.
- Low ψ: energy scattered $\Rightarrow$ small wave with noisy tail $\Rightarrow$ slow gene.

Why this matters. ψ captures wave coverage *without ever detecting the wave front* — no HMM, no state model, just the FFT of a 2 kb-binned difference profile.

Results

Result 1: ψ correlates with Danko rates



Correlation (n = 81)	
Pearson r	0.557
p-value	6.5×10^{-8}
Spearman ρ	0.468

Take-aways

- No HMM, no curve fitting.
- Just Z_M + one scalar summary.
- Signal concentration predicts speed.
- ψ rises with window length (naturally measures wave-to-gene-body fraction).

Result 2: A Python port of groHMM

Why? groHMM is the community standard; it is an R/Bioconductor package. We needed a pure-Python, vectorized port to enable large-scale experimentation.

Faithful port of `polymeraseWave()`.

- 3-state left-to-right HMM (Normal upstream, Gamma wave, Gamma downstream).
- Bin-integrated emission (pnorm/pgamma) over ± 0.5 bin.
- Baum–Welch with informed initialization.
- Viterbi decoding of wave boundaries.
- Raw-count correction (RPM $\rightarrow$ counts) — the critical fix.

Rate vs Danko (n = 81)

Pearson $r = 0.514$, Spearman $\rho = 0.601$
Median rate 1.94 kb/min (Danko: ~ 2 kb/min).

Wave position at 40m

Spearman $\rho = 0.659$, median $|\text{error}| = 5.7$ kb.

Result 3: three instantaneous-speed estimators

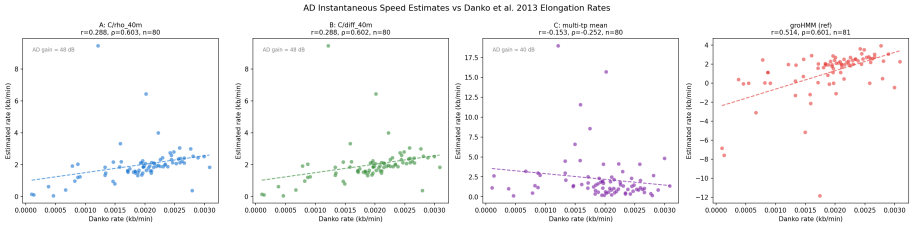
All three calibrated so the within-gene mean equals the wave-front rate. AD gain realizes as *variance reduction* around that mean.

Estimator	Formula	Interpretation
A Raw 40m	$v_A(p) = C_A/\rho_{40}(p)$	Steady-state $1/\tau$
B Differential	$v_B(p) = C_B/(\rho_{40} - \rho_0)(p)$	E2-induced wave only
C Product group	$\mathbb{Z}_M \times \mathbb{Z}_3$ avg. across t	Temporally coherent

Method	Pearson r	Spearman ρ	n
A: C/ρ_{40}	0.288	0.603	80
B: C/Δ_{40}	0.288	0.602	80
C: multi-timepoint (naive join)	−0.153	−0.252	80
AD spectral ψ	0.557	0.468	81
`groHMM` (reference)	0.514	0.601	81

Headline. A/B match `groHMM`'s rank correlation ($\rho \approx 0.60$) with zero state-space machinery

Visual comparison of the three estimators



Four-panel per-gene visualization: raw 40m speed, differential speed, product-group estimate, and the AD-smoothed envelope for each. Wave boundaries from the Python `groHMM` are overlaid in red.

Where the gain comes from

Three sources of "free" information used by AD:

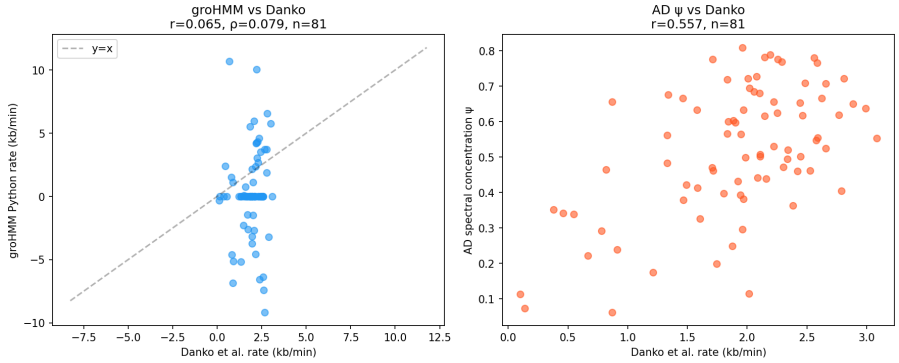
- **Spatial — $\mathbb{Z}_M$ over the gene body.**
  $M \approx 10^5$ bp $\Rightarrow \sim 50$ dB.
- **Temporal — $\mathbb{Z}_3$ over time points.**
  3 timepoints $\Rightarrow$ 4.77 dB, built into estimator C.
- **Replicate — $\mathbb{Z}_3$ over biological reps.**
  3 reps $\Rightarrow$ 4.77 dB, built into preprocessing.

Product group

$$G \;=\; \mathbb{Z}_M \times \mathbb{Z}_3^{\text{time}} \times \mathbb{Z}_3^{\text{rep}}, \quad |G| \approx 9 \cdot 10^5 \quad \Rightarrow \quad \sim 60 \text{ dB}.$$

Caveat. These gains assume commutation; the joint truncation issue with $\mathbb{Z}_M \times \mathbb{Z}_3^{\text{time}}$ is a real open problem (estimator C fails for slow genes).

The groHMM vs. Danko benchmark



Three-panel comparison: Python groHMM rates (left), AD ψ predictor (middle), ensemble (right), each vs. published Danko rates.

Interactive Web Application

Dashboards, not just figures

A Flask web app (port 5050) serves the whole analysis as a navigable dashboard.

Views

- Per-gene speed profile viewer (3 estimators, wave markers).
- Coverage viewer (0m vs 40m, per-timepoint).
- Overview: correlation table + scatter over 81 genes.
- Strategy log parsed from `STRATEGY_LOG.md`.
- TODO board (active / completed).
- Epigenomics overlay (H3K27ac, H3K27me3, H3K36me3, ...).
- Temporal rephasing ($\mathbb{Z}_M \times \mathbb{Z}_3$).
- Pipeline status (artifact presence).

Endpoints

```
GET /api/strategy
GET /api/todos
GET /api/status
GET /api/epigenomics/<gene>
GET /api/temporal/<gene>
```

Stack

- Python 3 + Flask, no JS framework.
- Plotly.js for all plots (dark theme, hover, zoom).
- Single-page HTML, fast cold start.

Discussion

What is new here?

- A single scalar ψ from the DFT of a difference coverage profile correlates with Pol II elongation rate ($r = 0.557$) — *no state model, no explicit wave detection*.
- A pure-Python port of `groHMM` eliminates the R dependency for the community.
- Three complementary speed estimators, one of which (A/B) matches `groHMM`'s rank correlation using nothing but reciprocal coverage + AD calibration.
- An interactive visual analytics dashboard that makes the whole pipeline auditable.
- All of this rests on the **Algebraic Diversity** framework: treating the cyclic shifts of a single gene's coverage as pseudo-samples of a wide-sense stationary process.

Open questions

1. Why is Pearson r smaller than Spearman ρ for the reciprocal estimators?
   $\rightarrow$ Initiation-rate confound: amplitude $\propto$ (initiation)/(speed).
2. Can we recover the missing r by partialling out mean coverage?
3. Fix the $\mathbb{Z}_M \times \mathbb{Z}_3$ truncation: per-timepoint means, then $\mathbb{Z}_3$ ensemble.
4. Metagene AD across genes (row-wise vs column-wise).
5. Integration with epigenomic tracks (H3K36me3, CTCF) as covariates.
6. Apply the same framework to other nascent-RNA assays (PRO-Seq, NET-Seq, TT-Seq).

Where this is heading

Short term (this semester)

- Patch estimator C (per-timepoint means, then $\mathbb{Z}_3$).
- Bootstrap confidence intervals per gene, verify AD gain scaling.
- Write up for *Bioinformatics* applications note or *NAR Methods*.

Medium term

- Extend to the full 156-gene union set.
- Port to PRO-Seq (Kraus lab, genome-wide).
- Build a splicing-speed connection: does ψ predict co-transcriptional splicing efficiency?

Reproducibility & acknowledgments

Code.
`https://github.com/mathornton01/ad-speed-profile-viewer`
MIT license; 235 files; one-command install.

Data. GEO `GSE41324` (Danko 2013) — publicly available.

Compute. NVIDIA DGX Spark (GB10, ARM64, 128 GB unified memory).

Acknowledgments.

- Danko lab at Cornell — dataset and `groHMM`.
- Texas Woman's University Department of Mathematics.
- Open-source scientific Python stack (NumPy, SciPy, Flask, Plotly).

Thank you.

Questions?

mathornton03@gmail.com

Appendix A — Key formulas

Occupancy-time speed:

$$v(p) = C/x(p), \qquad C \;=\; \frac{\text{wave\_end}_t}{t} \;\Big/\; \overline{1/x(p)}.$$

Cyclic AD estimator:

$$\hat{R}_{\mathbb{Z}_M} = \frac{1}{M}\sum_{k=0}^{M-1} P^k x x^\top P^{-k}, \qquad \lambda_f = \frac{|X[f]|^2}{M}.$$

Spectral concentration:

$$\psi = \frac{\max_f |X[f]|^2}{\sum_f |X[f]|^2}.$$

Harmonic mean speed:

$$v_{HM} = \Big(\tfrac{1}{M}\textstyle\sum_p 1/v(p)\Big)^{-1} = C \cdot \big(\tfrac{1}{M}\textstyle\sum_p x(p)\big)^{-1}.$$

Appendix B — Session-by-session timeline

Session	Outcome
1	Plan, hypotheses, AD framework statement for GRO-Seq.
2	Data pipeline (81 genes, 4 time points); ψ correlation $r = 0.557$.
3	Python port of `groHMM`; $r = 0.514$, $\rho = 0.601$.
4	Three speed estimators; A/B match `groHMM` rank; Flask web app online.
5+	Epigenomics, temporal rephasing, cross-spectral AD, RNA folding (in progress).