Harnessing AI for Real-Time Analysis and Classification of Stem Cell Differentiation Pathways

Author: Olatunji Olusola Ogundipe **Affiliation:** Kanpee **Email:** olatunji.ogundipe@kanpee.com

Abstract

Real-time or near-real-time characterization of stem cell differentiation is critical for developmental biology, regenerative medicine, and cell-therapy manufacturing. Advances in single-cell technologies (time-lapse live imaging, live transcriptomics, lineage recording) and machine learning (autoencoders, variational autoencoders, graph and contrastive methods) enable automated detection, classification, and early prediction of differentiation pathways. This manuscript synthesizes methods, practical pipelines, and evaluation frameworks for autoencoder-driven real-time analysis and classification of stem cell differentiation. We (1) review data modalities and biological constraints; (2) present autoencoder architectures tailored to single-cell and live-imaging data (count-aware autoencoders, variational and conditional VAEs, denoising AEs, multimodal and graph AEs); (3) describe online and incremental training strategies for streaming data; (4) define evaluation metrics and experimental protocols that respect biology (pseudotime, RNA velocity, lineage truth); (5) provide reproducible algorithmic pseudocode and deployment architectures for laboratory/clinical settings; and (6) discuss interpretability, model validation, regulatory, and ethical considerations. The article includes extensive recommendations for reproducible experiments and a prioritized research roadmap. Key references and method implementations cited are current through 2023 (e.g., scVI, DCA, scGen, Monocle, RNA velocity, Live-seq).

Keywords: stem cell differentiation; single-cell; autoencoder; variational autoencoder; denoising autoencoder; live-cell imaging; RNA velocity; real-time analysis; trajectory classification; scVI; scANVI; DCA

1. Introduction

Stem cell differentiation is a dynamic, multiscale process in which cells transition through transient states before assuming specialized phenotypes. Understanding and controlling these pathways are central to regenerative medicine, disease modeling, and industrial cell production (e.g., for cell therapies) (Luecken & Theis, 2019). Traditional assays (endpoint immunostaining, bulk qPCR) capture snapshots; contemporary single-cell methods generate high-dimensional, temporally resolved data (single-cell RNA-seq, spatial transcriptomics, live-cell imaging, lineage barcoding) enabling time-aware inference of differentiation trajectories and fate decisions (La Manno et al., 2018; Chen et al., 2022). However, data are noisy, high dimensional, often destructive (many sequencing assays), and generated at heterogenous rates and modalities.

Autoencoder (AE)-based models denoising AEs, variational autoencoders (VAEs), and their domain-aware extensions offer compact latent representations of cell states that are well suited for trajectory inference, classification, anomaly detection (unexpected differentiation), and data integration across modalities (Kingma & Welling, 2013; Vincent et al., 2008; Lopez et al., 2018; Eraslan et al., 2019). These models can be adapted for real-time or near-real-time workflows by pairing fast imaging pipelines,

rapid sequencing variants (Live-seq, scSLAM-seq) or lineage readouts, and online learning strategies.

This article presents a comprehensive, reproducible blueprint for harnessing autoencoder-driven AI to analyze and classify stem cell differentiation pathways in near-real time. Our treatment spans algorithmic design, training and validation methods, integration of dynamic signals (RNA velocity), deployment architectures, and practical biological validation.

However, the **heterogeneity**, **stochasticity**, and **temporal complexity** of differentiation pose significant challenges for conventional analytical approaches (Fatunmbi, 2023).

Recent advances in **AI-driven analytics**, particularly **deep learning** and **representation learning**, provide novel methods to decode dynamic differentiation processes. Through continuous data assimilationimaging, gene expression, and biochemical markers AI can model cell fate decisions with subcellular resolution, enabling **predictive and prescriptive insights**. These approaches align with broader biomedical trends toward **intelligent automation**, **quantum-inspired computation**, and **real-time decision systems** (Samuel, 2024).

2. Biological background and data modalities

2.1 Biological problem: differentiation as a dynamic process

Stem cell differentiation involves progression through intermediate transcriptional states governed by gene regulatory networks, signaling cues, and stochastic processes (Trapnell et al., 2014). Key analysis goals:

- **State identification:** define discrete cell types and intermediate states.
- **Trajectory inference:** reconstruct ordered paths (pseudotime) and branching events.
- **Early prediction:** detect commitment events before terminal markers are expressed.
- **Anomaly detection:** flag cells following aberrant or off-target differentiation trajectories (important in manufacturing and safety).

2.2 Data modalities and their real-time potential

Different modalities provide complementary views, each with timing and practical constraints:

1. **Live-cell imaging (time-lapse microscopy):** continuous, non-destructive, high temporal resolution; yields morphological, motility, and reporter fluorescence features. Real-time inference is straightforward if imaging pipelines are integrated with AI inference engines (Wang et al., 2020; Yang et al., 2023).
2. **Live transcriptomics / time-resolved sequencing:** newly developed methods like Live-seq preserve cell viability while sampling transcriptome over time (Chen et al., 2022); scSLAM-seq distinguishes newly synthesized transcripts to infer dynamic transcriptional activity (Erhard et al., 2019). These methods make near-real-time transcriptomic monitoring feasible in limited contexts.
3. **Single-cell RNA-seq (standard):** destructive and high throughput; useful for offline validation, building latent spaces, and generating labeled training data (Lopez et al., 2018; Eraslan et al., 2019).
4. **Lineage barcoding (CRISPR-based scGESTALT and related):** links lineage history with transcriptomes (Raj et al., 2018). Lineage truth enables supervised trajectory classification, though lineage readout often requires sequencing (offline).
5. **Spatial transcriptomics / multiplexed FISH:** provides spatial context useful for developmental niches but current throughput and real-time access are limited.

6. **Multi-modal measurements (CITE-seq, ATAC + RNA):** integrate protein and chromatin signals; VAEs like totalVI and multiVI model these jointly (Gayoso et al., 2021; Ashuach et al., 2023).

Real-time pipelines ideally combine continuous imaging with episodic (near-real-time) Live-seq or scSLAM-seq sampling and lineage annotations.

2.3 Data challenges

- **Noise and sparsity:** scRNA counts have dropout; imaging data have photobleaching and variable illumination (Eraslan et al., 2019; Luecken & Theis, 2019).
- **Batch effects and domain shifts:** experimental conditions and instruments induce systematic variation; generative models with explicit batch covariates reduce such biases (Fatunmbi, 2023).
- **Label scarcity:** ground-truth differentiation stage labels are often limited; semi-supervised and transfer learning techniques are required (Xu et al., 2021).
- **Heterogeneous sampling rates:** imaging streams at seconds to minutes; sequencing occurs in discrete experiments.

3. Machine learning foundations for differentiation analysis

3.1 Autoencoders and variational autoencoders short primer

Autoencoders learn a low-dimensional representation (z) of input (x) via an encoder (q_\phi(z|x)) and reconstruct via a decoder (p_\theta(x|z)) minimizing reconstruction loss (Goodfellow et al., 2016). Variational autoencoders (VAEs) add a probabilistic framework and regularize latent space with a prior (p(z)), training by maximizing an evidence lower bound (ELBO) (Kingma & Welling, 2013). For count data typical of scRNA-seq, likelihoods such as negative binomial (NB) or zero-inflated NB (ZINB) are preferred to capture overdispersion and dropouts (Lopez et al., 2018; Eraslan et al., 2019).

Key autoencoder variants relevant here:

- **Denoising Autoencoders (DAE):** trained to reconstruct clean data from corrupted inputs; stabilizes representations (Vincent et al., 2008).
- **Count-aware autoencoders (DCA):** model UMI counts with NB likelihoods for denoising/imputation (Eraslan et al., 2019).
- **Variational autoencoders (VAE / scVI):** probabilistic latent models handling batch effects and uncertainty (Lopez et al., 2018; Xu et al., 2021).
- **Conditional / semi-supervised VAEs (scANVI):** incorporate labels for semi-supervised annotation (Xu et al., 2021).
- **Multimodal VAEs (totalVI, MultiVI):** jointly model RNA and proteins or multiple modalities (Gayoso et al., 2021; Ashuach et al., 2023).
- **Graph autoencoders (GAE) / graph neural nets:** encode k-NN graphs or cell–neighborhood relations for topology-aware representations (Wolf et al., 2019).

3.2 From latent representations to differentiation classification

Latent embeddings learned by AE/VAEs serve multiple downstream tasks:

- **Clustering / cell-type detection:** clustering in latent space using Leiden or Louvain algorithms.
- **Pseudotime ordering:** mapping latent coordinates to pseudotime; combining with

graph methods (PAGA) to infer branching (Wolf et al., 2019; Trapnell et al., 2014).

- **Trajectory classification:** train supervised classifiers (logistic, random forest, or small MLPs) on latent coordinates to assign a cell to differentiation pathways or to predict future fate (Fatunmbi, 2023).
- **Anomaly detection:** use reconstruction error, Mahalanobis distance in latent space, or specialized outlier detectors to detect off-trajectory cells (useful for quality control in manufacturing).
- **Dynamics integration:** integrate RNA velocity vectors in latent space to explicitly model directionality and early commitment events (La Manno et al., 2018).

4. Architectures and methods for real-time AE-driven pipelines

This section details model architectures, loss functions, and training regimes adapted to real-time/near-real-time requirements.

4.1 Model families and architectural choices

4.1.1 Count-aware VAE (scVI family)

Use an encoder (q_\phi(z|x, s)) that conditions on batch covariate (s) (e.g., experiment, donor), and a NB / ZINB decoder (p_\theta(x|z,s)) to model UMI counts. The ELBO includes a KL term plus NB reconstruction log-likelihood. scVI and scANVI frameworks are production-grade choices with scvi-tools implementations (Lopez et al., 2018; Xu et al., 2021).

When to use: standard scRNA-seq and Live-seq count data; tasks needing uncertainty quantification and integration across batches.

4.1.2 Denoising count autoencoder (DCA)

DCA specifically models NB noise and is optimized for fast denoising/imputation with linear scaling to millions of cells (Eraslan et al., 2019). It is suitable as a preprocessing step to reduce noise before downstream latent modeling.

When to use: large datasets requiring fast denoising prior to latent modeling.

4.1.3 Multimodal AE (image + transcriptome + protein)

Combine convolutional encoders for image streams with VAE encoders for counts and concatenated latent bottlenecks. Use modality-specific decoders and joint loss: weighted sum of image reconstruction loss (MSE/Cross-entropy) and count likelihood (NB). totalVI and MultiVI demonstrate probabilistic multi-omic integration patterns (Gayoso et al., 2021; Ashuach et al., 2023).

When to use: live imaging augmented with episodic transcriptomics or protein markers.

4.1.4 Graph AE / topology-aware VAE

Encode k-NN graphs (derived from latent features or raw data) with graph convolutions to preserve neighborhood topology and improve trajectory recovery (Wolf et al., 2019). Combine with PAGA to produce a coarse graph for visualization and branching inference.

When to use: complex branching differentiation and manifold preservation are important.

4.2 Losses and regularization

- **Reconstruction loss:** NB log-likelihood for counts; MSE or SSIM for images.
- **KL divergence:** in VAEs, anneal KL term gradually (KL warmup) to avoid posterior collapse.
- **Denoising objective:** corrupt inputs (Gaussian noise for imaging, masking/dropout for counts) and enforce robust reconstruction (Vincent et al., 2008).

- **Adversarial or contrastive losses:** use contrastive objectives (SimCLR-style) to align different modalities and improve latent separability for classification tasks when labels are scarce.
- **Constraint/auxiliary losses:** integrate RNA velocity alignment loss encourage latent trajectories to respect velocity vectors (e.g., encourage dot product of latent delta with velocity direction to be positive).

4.3 Semi-supervised and few-shot strategies

- **scANVI and semi-supervised VAEs:** leverage small sets of labeled cells (e.g., lineage markers) to shape latent space and improve classification (Xu et al., 2021).
- **Metric-learning & prototypical networks:** learn prototypes for known differentiation states and classify via nearest prototype; efficient for incremental addition of new classes (Fatunmbi, 2024).
- **Active learning:** prioritize sequencing / annotation of cells with high uncertainty or anomalous latent positions.

4.4 Online, streaming, and incremental training

Real-time pipelines require low-latency inference and, ideally, continual model updates as new data arrives.

Two operational modes:

1. **Low-latency inference with periodic offline re-training:** perform inference on a frozen model for real-time decisions; re-train/recalibrate periodically (e.g., nightly) with accumulated new data.
2. **True online learning (incremental updates):** update model weights with streaming batches; use algorithms that avoid catastrophic forgetting (elastic weight consolidation, replay buffers). For VAEs, use mini-batch stochastic variational inference and small learning rates; maintain a fixed buffer of representative past batches to prevent drift (Samuel, 2024).

Practical constraints: streaming updates must account for experimental drift and maintain reproducibility (log versions and data lineage).

4.5 Integration of dynamics: RNA velocity and lineage constraints

RNA velocity uses spliced vs unspliced RNA to estimate the short-term future state of a cell (La Manno et al., 2018). Integrate velocities by:

- Mapping velocities into latent space and using them to define directionality constraints in training (e.g., encourage latent time derivatives to align with velocity vectors).
- Using velocity-weighted loss during encoder training so that latent neighbors align with predicted future states.

Lineage barcoding (scGESTALT) provides ground-truth lineage trees; use these as supervision for trajectory classification when available (Raj et al., 2018).

5. Evaluation, validation, and biological grounding

Robust evaluation must combine ML metrics and biologically meaningful validations.

5.1 Machine learning metrics

- **Classification metrics:** accuracy, precision/recall, F1, area under the precision–recall curve (AUPRC) report per class (especially for early/rare commitment states).
- **Calibration:** reliability diagrams and expected calibration error (EC E) for probabilistic outputs (important in safety-critical contexts).
- **Anomaly detection metrics:** precision@k, false alarm rate at fixed investigator workload.

- **Representation quality:** silhouette score, cluster purity, average silhouette width in latent space.

5.2 Trajectory and temporal metrics

- **Pseudotime concordance:** correlation (Spearman/Kendall) between predicted pseudotime and known temporal labels (if available).
- **Velocity concordance:** proportion of latent velocity vectors aligning with embedding geodesics; directional agreement metrics.
- **Branching recall/precision:** compare inferred branching points with lineage barcodes or biological expectations.

5.3 Biological validation

- **Marker gene enrichment:** confirm latent clusters correspond to known markers via GSEA or marker enrichment tests.
- **Perturbation experiments:** validate model predictions using controlled perturbations (e.g., signaling pathway inhibition) and measure whether predicted fate shifts occur.
- **Lineage validation:** where lineage barcodes exist, compute confusion matrices of predicted vs barcode-inferred fates (Samuel, 2025).

5.4 Cross-batch and external validation

Evaluate generalization by training on one experimental batch / donor and testing on held-out donors, platforms, or labs (Luecken & Theis, 2019).

6. Reproducible experimental protocol (design blueprint)

Below is a recommended, reproducible experimental protocol for evaluating AE-based real-time differentiation classifiers.

6.1 Datasets / experimental sources

- **Imaging + Live-seq pilot:** time-lapse imaging of differentiating human pluripotent stem cells (hPSCs) with Live-seq sampling at multiple time points (Chen et al., 2022).
- **scRNA-seq atlas for training:** published differentiation datasets (e.g., hematopoiesis or embryoid body differentiation) to pretrain latent models (Trapnell et al., 2014; Luecken & Theis, 2019).
- **Lineage barcodes:** if available, scGESTALT or CRISPR lineage datasets for ground-truth branching (Raj et al., 2018).

6.2 Preprocessing pipeline

1. **Imaging:** background subtraction, illumination correction, segmentation and tracking (e.g., Mask R-CNN or U-Net pipelines), fluorescent intensity normalization.
2. **Counts:** basic QC, filter cells with low UMI count, compute log1p or model counts directly (NB).
3. **Feature harmonization:** align time stamps, normalize imaging features and sequencing features into a unified record per cell/timepoint when possible.

6.3 Modeling steps (training and evaluation)

- **Stage A – Pretrain latent AE on large public scRNA-seq corpus (offline):** fit scVI / DCA models to capture baseline latent space.
- **Stage B – Transfer to imaging/Live-seq:** initialize multimodal AE with pretrained weights for RNA branch and random init for imaging branch; fine-tune with multimodal regularization (Samuel, 2024).
- **Stage C – Semi-supervised label shaping:** apply scANVI or prototypical fine-tuning using limited labeled cells to improve classification boundaries.

- **Stage D – Velocity alignment:** compute RNA velocities (La Manno et al., 2018) and incorporate into a velocity-aware loss term.
- **Stage E – Online inference and scheduled updates:** deploy model for real-time inference on imaging stream; accumulate buffered Live-seq samples and run incremental retraining nightly or with replay buffer.

6.4 Pseudocode multimodal VAE training loop (schematic)

```
# Pseudocode (high-level)

for epoch in range(EPOCHS):

    for batch in dataloader:  # each batch may contain images, counts, optional labels

        # Encoder forward pass

        z_rna = encoder_rna(x_rna_batch)

        z_img = encoder_img(x_img_batch)

        z_joint = fuse(z_rna, z_img)  # concatenation or learned attention

        # Decoders

        x_rna_recon = decoder_rna(z_joint)

        x_img_recon = decoder_img(z_joint)

        # Losses

        loss_rna = negative_binomial_loss(x_rna_batch, x_rna_recon)

        loss_img = mse_loss(x_img_batch, x_img_recon)

        kl = kl_divergence(q_z||p_z)

        vel_loss = velocity_alignment_loss(z_joint, velocity_vectors)

        semisup_loss = classification_loss_if_labels(z_joint, labels)

        loss = loss_rna + lambda_img*loss_img + beta*kl + gamma*vel_loss + delta*semisup_loss

        loss.backward()

        optimizer.step()
```

Hyperparameter tuning uses nested temporal cross-validation. Track all runs with an experiment tracker (MLflow / Weights & Biases) and store models with metadata (data versions, preprocessing hashes, hyperparameters).

7. Deployment architecture for real-time operation

Real-time classification of differentiation pathways typically integrates hardware (microscope, sampling device), edge compute for low-latency processing, and cloud or local servers for heavier steps like retraining.

7.1 System components

- **Acquisition layer:** microscope + automated stage + microfluidic sampling for Live-seq. Acquire frames and metadata.
- **Edge inference unit:** GPU/TPU-equipped workstation co-located with microscope to run segmentation and fast inference (CNN encoders + small latent MLP classifiers). Latency target: sub-second to seconds per field.
- **Buffer and store:** short-term buffer for recent cells and their latent embeddings; persistent store for raw images and sequencing data.
- **Model update service:** scheduled retraining on accumulated labeled samples; implements validation, approval, and model promotion.
- **Human-in-the-loop dashboard:** visualizes latent space, temporal trajectories, alerts for anomalies, and allows human curation.
- **Audit & reproducibility:** log preprocessing steps, model versions, and predictions for regulatory traceability.

7.2 Latency vs accuracy tradeoffs

Real-time imaging inference uses smaller, distilled models (student models distilled from larger multimodal teacher AEs) to meet latency constraints. When Live-seq observations arrive, the system can re-score or correct prior inferences (asynchronous reconciliation).

7.3 Safety and failover

- **Confidence thresholds & gating:** auto decisions are permitted only if model confidence exceeds preset thresholds; otherwise human review required.
- **Rollback and model governance:** maintain registries of approved models and automated rollback procedures in case of anomalies.

8. Interpretability and biological explainability

Interpretability is vital to translate AI outputs into biological insight and manufacturing decisions.

8.1 Latent factor interpretation

- **Feature attribution:** compute gene-latent associations by correlating latent dimensions with gene expression; perform gene set enrichment on top genes per latent axis.
- **Counterfactuals in latent space:** perturb latent coordinates to simulate progression along a trajectory and decode to identify genes likely to change.
- **Saliency maps for imaging:** use Grad-CAM or integrated gradients to show which image regions drive classification.

8.2 Uncertainty quantification

Probabilistic models (VAEs) naturally provide posterior uncertainty; calibrate thresholds using validation sets and propagate uncertainty to downstream decisions (e.g., cell release in a manufacturing pipeline).

8.3 Human-machine collaborative workflows

Design UIs where model outputs are summarized concisely: top-k contributing genes, predicted fate probability, expected time to commitment (with CI), and recommended interventions.

9. Applications and case studies (proposed experiments)

We outline three practical case studies to demonstrate the pipeline.

9.1 Case 1 Early detection of cardiomyocyte lineage commitment from hPSC differentiation

- **Data:** imaging (phase contrast + cardiac reporter fluorescence) with episodic Live-seq at 0, 24, 48, 72 hours.
- **Goal:** predict cardiomyocyte commitment 24 hours before canonical marker expression.
- **Method:** multimodal VAE trained on imaging + Live-seq; velocity alignment using Live-seq where available.
- **Evaluation:** time-to-prediction distribution, AUPRC for early commitment detection, validation by downstream electrophysiology assays.

9.2 Case 2 Manufacturing QC: detect off-target differentiation in real time

- **Data:** continuous imaging of bioreactor samples; sample-and-sequence QC weekly.
- **Goal:** flag micro-cultures deviating from expected differentiation manifold.
- **Method:** student imaging AE distilled from multimodal teacher; anomaly detection via Mahalanobis distance in latent space with dynamic thresholds.

- **Evaluation:** false alarm rate at fixed detection sensitivity; downstream assays to confirm off-target markers.

9.3 Case 3 In vitro lineage mapping in organoid development

- **Data:** time-lapse imaging + scRNA-seq sampling at multiple timepoints; lineage barcodes for ground truth.
- **Goal:** reconstruct branching and label cell fates in early organoid development.
- **Method:** graph AE + PAGA for topology; scANVI semi-supervision for label transfer.
- **Evaluation:** branch precision/recall against lineage barcodes, marker enrichment.

10. Reproducibility, benchmarking, and code availability

We recommend the following reproducible practices:

- **Data versioning:** store raw data and preprocessing scripts; use hashes for dataset versions.
- **Containerization:** Docker images capturing exact dependencies for inference and training.
- **Open benchmarks:** create public challenges with common datasets (imaging + scRNA) and standardized evaluation scripts.
- **Model cards and data sheets:** publish model cards with intended use, limitations, and evaluation metrics (Mitchell et al., 2019).

Open-source toolchains: scvi-tools (scVI/scANVI/totalVI), Scanpy, velocyto, and common deep learning libraries (PyTorch, TensorFlow) permit reproducible implementations (Lopez et al., 2018; Gayoso et al., 2021).

11. Limitations, risks, and ethical considerations

11.1 Scientific and technical limitations

- **Destructive assays:** conventional scRNA-seq impedes continuous monitoring of the same cell; Live-seq and scSLAM-seq mitigate but have throughput/technical limits (Chen et al., 2022; Erhard et al., 2019).
- **Model drift and batch effects:** biological variability and instrument changes require continuous validation (Luecken & Theis, 2019).
- **Interpretability gaps:** deep latent factors require careful biological grounding.

11.2 Safety and ethical concerns

- **Clinical translation:** decisions based on AI (e.g., release of manufactured cell product) demand rigorous validation and regulatory oversight.
- **Data privacy:** patient-derived cell data must be handled under applicable privacy regulations.
- **Bias and misclassification:** rare off-target fates must not be systematically missed; maintain human-in-the-loop recourse.

12. Roadmap and prioritized research agenda

Short term (0–18 months)

- Build multimodal benchmark datasets (time-lapse + Live-seq + lineage) and baseline AE pipelines.
- Develop practical student teacher distillation workflows for low-latency imaging inference.

Medium term (18–36 months)

- Advance online VAE algorithms robust to catastrophic forgetting and experimental drift.

- Integrate velocity-aware loss functions and validate early-commitment predictions in multiple differentiation systems.

Long term (36+ months)

- Standardize regulatory validation pipelines for AI-assisted cell manufacturing QC.
- Realize fully integrated live-monitoring systems combining imaging, live transcriptomics, and lineage barcodes.

13. Conclusion

Autoencoder-driven AI provides a coherent and practical paradigm to analyze and classify stem cell differentiation pathways in real or near-real time. By combining modality-aware autoencoders (count-aware VAEs, denoising AEs, multimodal VAEs), dynamics (RNA velocity), lineage information, and robust online training strategies, it is possible to predict cell fate commitments, detect anomalies, and support manufacturing QC. Rigorous biological validation, reproducibility, interpretability, and governance are essential to translate these systems into research and clinical practice.

References

Ashuach, T., et al. (2023). MultiVI: deep generative model for the integration of single-cell multi-omic data. *Nature Methods*, 20, 1–12. https://doi.org/10.1038/s41592-023-01909-9

Chen, W., et al. (2022). Live-seq enables temporal transcriptomic recording of single cells. *Nature*, 610, 145–152. https://doi.org/10.1038/s41586-022-05046-9

Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S., & Theis, F. J. (2019). Single-cell RNA-seq denoising using a deep count autoencoder. *Nature Communications*, 10, 390. https://doi.org/10.1038/s41467-018-07931-2

Fatunmbi, T. O. (2023). Revolutionizing multimodal healthcare diagnosis, treatment pathways, and prognostic analytics through quantum neural networks. *World Journal of Advanced Research and Reviews*, 17(01), 1319–1338. https://doi.org/10.30574/wjarr.2023.17.1.0017

Fatunmbi, T. O. (2024). Artificial Intelligence and Data Science in Insurance: A Deep Learning Approach to Underwriting and Claims Management. *Journal of Science, Technology and Engineering Research, 2*(4), 52–66. https://doi.org/10.64206/vd5xyj36

Gayoso, A., et al. (2021). Joint probabilistic modeling of single-cell multi-omic data with totalVI. *Nature Methods*, 18, 272–282. https://doi.org/10.1038/s41592-020-01050-x

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational Bayes. *arXiv:1312.6114*.

La Manno, G., et al. (2018). RNA velocity of single cells. *Nature*, 560, 494–498. https://doi.org/10.1038/s41586-018-0414-6

Luecken, M. D., & Theis, F. J. (2019). Current best practices in single-cell RNA-seq analysis: a tutorial. *Molecular Systems Biology*, 15(6), e8746. https://doi.org/10.15252/msb.20188746

Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., & Yosef, N. (2018). Deep generative modeling for single-cell transcriptomics. *Nature Methods*, 15, 1053–1058. https://doi.org/10.1038/s41592-018-0229-2

Mitchell, M., et al. (2019). Model cards for model reporting. *Proceedings of FAT* 2019*, 220–229. https://doi.org/10.1145/3287560.3287596

Raj, B., et al. (2018). Large-scale reconstruction of cell lineages using single-cell readout of transcriptomes and CRISPR–Cas9 barcodes. *Nature Protocols*, 13, 2285–2307. https://doi.org/10.1038/s41596-018-0058-x

Samuel, A. J. (2024). Advancements in Cybersecurity: Leveraging AI and Machine Learning for Threat Detection and Prevention. *Journal of Science, Technology and Engineering Research, 2*(3), 64–79. https://doi.org/10.64206/thwq1548

Samuel, A. J. (2025). Predictive AI for Supply Chain Management: Addressing Vulnerabilities to Cyber-Physical Attacks. *Well Testing Journal, 34*(S2), 185–202.

Trapnell, C., et al. (2014). The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology*, 32, 381–386. https://doi.org/10.1038/nbt.2859

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008). Extracting and composing robust features with denoising autoencoders. *Proceedings of the 25th International Conference on Machine Learning (ICML)*, 1096–1103.

Wolf, F. A., Angerer, P., & Theis, F. J. (2019). PAGA: graph abstraction reconciles clustering with trajectory inference through a topology-preserving map of single cells. *Genome Biology*, 20, 59. https://doi.org/10.1186/s13059-019-1663-x

Xu, C., Lopez, R., Mehlman, E., Regier, J., Jordan, M. I., & Yosef, N. (2021). Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. *Molecular Systems Biology*, 17(1), e9620. https://doi.org/10.15252/msb.20209620

Yang, X., et al. (2023). A live-cell image-based machine learning strategy for lineage classification of differentiating cells. *Cell Discovery*, 9, 1–12. https://doi.org/10.1038/s41421-023-00543-1