Quantum Machine Learning for Genomic Data Analysis: Unlocking Precision Medicine via Hybrid AI Systems

Author: Matthew Harris, **Affiliation:** Lecturer, Faculty of AI and Robotics, University of Helsinki, Finland.

Email: matthew.harris@helsinki.fi

Abstract

Genomic data characterized by extreme dimensionality, complex structure, and subtle signal-to-noise ratios presents formidable computational and statistical challenges for precision medicine. Quantum machine learning (QML) and hybrid quantum–classical AI systems offer novel computational paradigms that may accelerate or improve aspects of genomic analysis: from sequence alignment and assembly to variant calling, haplotype phasing, population genetics, and multi-omics integration. This article develops a comprehensive, scholarly account of QML applied to genomics. We (i) review the theoretical foundations of quantum algorithms and QML models relevant to genomics (quantum kernels/feature maps, variational quantum circuits, quantum annealing); (ii) formalize problem mappings for core genomics tasks and provide explicit encodings (k-mer embeddings, binary/angle encodings, QUBO formulations); (iii) propose hybrid AI system architectures combining classical deep learning with quantum subroutines for discrete and high-dimensional subproblems; (iv) delineate experimental protocols, benchmarking strategies, and evaluation metrics that fairly compare QML to classical baselines; and (v) critically assess practical limitations (noise, scalability, data-encoding overhead), ethical and security implications, and a realistic roadmap for translational research. We ground our discussion with recent empirical findings and systematic reviews that evaluate QML's promise and limits in biological data domains. While existing quantum hardware is in the NISQ (noisy intermediate-scale quantum) era, hybrid approaches where quantum processors solve discrete combinatorial or kernel-evaluation subproblems inside largely classical pipelines present a pragmatic path toward early utility in genomics. We conclude with concrete recommendations for researchers and practitioners seeking to responsibly explore QML for precision medicine.

Keywords: quantum machine learning, genomics, hybrid AI, variational quantum circuits, quantum kernels, QUBO, quantum annealing, precision medicine

1. Introduction

The past two decades have witnessed an explosion of genomic data: whole-genome and whole-exome sequencing, large cohort studies, single-cell genomics, and multi-omics consortia create datasets of increasing size and complexity. Classical computational genomics has advanced rapidly leveraging algorithmic innovations, high-performance computing, and deep learning but fundamental computational bottlenecks remain. Tasks such as de novo assembly, haplotype phasing, exhaustive pairwise-similarity searches, and combinatorial optimization for pan-genome graph construction can be computationally intensive and, in some formulations, asymptotically intractable. Quantum computing proposes different asymptotic scaling for certain problems and provides new algorithmic primitives (superposition, entanglement, amplitude amplification) that can be exploited by hybrid algorithms. The central question that motivates this paper is straightforward but multifaceted: *Can quantum machine learning (QML) and hybrid quantum–classical systems materially improve*

genomic data analysis and thereby accelerate precision medicine?

The literature provides both optimism and caution. Landmark QML demonstrations such as quantum-enhanced feature spaces (quantum kernel methods) and variational quantum classifiers have shown proof-of-concept advantages on tailored problems and small datasets, exploiting a quantum state space as an exponentially large feature map that may be inaccessible classically. At the same time, systematic reviews and theoretical analyses highlight important limits, including embedding overheads, noise susceptibility, and that realistic quantum speedups often require careful problem structure and hardware advances.

This article maps the current landscape onto concrete genomic use cases, details how to encode biological data for quantum processing, proposes hybrid architectures that focus quantum resources where they are most likely to help (discrete optimization, kernel evaluations, sampling-heavy subroutines), and articulates rigorous experimental protocols for fair benchmarking. We emphasize practical translational pathways that account for the current NISQ ecosystem while remaining attentive to fault-tolerant futures.

Structure of the paper: Section 2 reviews genomics problem classes and computational challenges. Section 3 introduces quantum computing and QML primitives. Section 4 presents mappings of genomics tasks to quantum/hybrid formulations and gives algorithmic blueprints. Section 5 proposes hybrid system architectures and training/deployment patterns. Section 6 lays out experimental design and benchmarking strategies. Section 7 discusses practical limitations, security, and ethical considerations. Section 8 concludes with a roadmap for research and translation.

2. Genomic Data: Tasks, Properties, and Computational Challenges

2.1 Typical genomic data modalities and tasks

Genomic and related biomedical datasets include, but are not limited to:

- **Short-read and long-read sequencing reads** (Illumina, PacBio, Oxford Nanopore) used for assembly and variant discovery.
- **Variant callsets** (SNPs, indels, structural variants) and genotypes across cohorts.
- **Haplotype and phasing data** to reconstruct chromosome-scale allelic configurations.
- **Expression and epigenomic profiles** from bulk and single-cell assays.
- **Population genetic summaries** (linkage disequilibrium matrices, coalescent trees).
- **Sequence graphs / pangenomes** representing structural variation across populations.

Key computational tasks: sequence alignment, de novo assembly (graph construction & path finding), variant calling and genotyping, haplotype phasing, read error correction, motif finding, similarity search (e.g., k-mer matching, nearest-neighbor queries), population structure inference, and combinatorial optimization for experimental design (e.g., optimal primer design), among others.

2.2 Structural characteristics impinging algorithm design

Two features distinguish genomic data from many standard machine-learning datasets:

1. **Extreme dimensionality with structured sparsity.** Genomes involve sequences over alphabet $\Sigma=\{A,C,G,T\}$ yielding combinatorial explosion of k-mers; yet biological variation is sparse in meaningful dimensions (most positions conserved).
2. **Multi-scale structure.** Biological signals interact across nucleotides, motifs, genes, and chromosomal scales; this favors hierarchical and graph-based models.

3. **Discrete combinatorial optimization cores.** Problems such as haplotype phasing, assembly scaffolding, and read-ordering often reduce to NP-hard formulations (e.g., traveling-salesman-like ordering, minimum-error correction).
4. **Noise and error models.** Sequencing technologies introduce distinct error signatures (indel-heavy vs substitution errors) that affect algorithmic choices.

2.3 Computational bottlenecks and where quantum methods might help

Classical genomics benefits from algorithmic advances (FM-indexing, Burrows-Wheeler transform, succinct de Bruijn graphs) and scalability engineering. Yet practical bottlenecks remain:

- **All-pairs similarity** (e.g., large k-mer comparisons) scales poorly with cohort size.
- **Combinatorial optimization** (e.g., optimal set cover for annotation, haplotype assembly) can be resource-intensive.
- **Sampling from complex posterior distributions** (e.g., coalescent-based inference) may require heavy MCMC runs.

Quantum algorithms may eventually offer benefits in two broad ways: (i) *quantum-enhanced feature transformations* (quantum kernels) that can separate classes in high-dimensional quantum state spaces more effectively than classical kernels for certain data encodings; and (ii) *quantum solvers for combinatorial optimization* (quantum annealers, QAOA) that can serve as accelerators for discrete subproblems embedded within larger classical pipelines. However, gain is problem-dependent and must be empirically and theoretically validated.

3. Quantum Computing and Quantum Machine Learning: Principles and Primitives

This section provides a compact primer necessary to understand QML applications to genomics.

3.1 Quantum computation basics (brief)

A quantum bit (qubit) is a two-level quantum system represented as a normalized superposition $|\psi\rangle=\alpha|0\rangle+\beta|1\rangle$ $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ $|\psi\rangle=\alpha|0\rangle+\beta|1\rangle$. Quantum computation manipulates qubits by unitary gates and extracts classical outcomes via projective measurement. Key resources include **superposition** (representing 2n2^n2n amplitudes on nnn qubits), **entanglement** (non-classical correlations), and **interference**.

Current devices lie in the NISQ regime: dozens–low hundreds of noisy qubits, limited coherence times, gate errors, and constrained qubit connectivity. This strongly motivates hybrid algorithms that use short-depth quantum circuits inside classical workflows.

3.2 QML primitives relevant to genomics

3.2.1 Quantum feature maps and quantum kernels

A foundational idea in several QML approaches is to map classical data x∈Rdx\in\mathbb{R}^dx∈Rd to quantum states $|\phi(x)\rangle|\phi(x)\rangle|\phi(x)\rangle$ using parameterized circuits (feature maps). The overlap kernel $K(x,x')=|\langle\phi(x)|\phi(x')\rangle|2K(x,x') = |\langle\phi(x)|\phi(x')\rangle|^2K(x,x')=|\langle\phi(x)|\phi(x')\rangle|2$ can serve as the inner product in the induced (possibly exponentially large) quantum feature space. Havlíček et al. (2019) experimentally showcased that supervised learning using quantum feature maps can construct classifiers where the kernel is efficiently accessible on quantum hardware even when classical evaluation is hard, presenting a possible route to quantum advantage in specific learning problems.

3.2.2 Variational Quantum Circuits (VQCs) / Parameterized Quantum Circuits (PQCs)

VQCs are parameterized quantum circuits trained via a classical optimizer to minimize task loss (e.g., classification error). They are the defacto NISQ approach for supervised learning and generative modeling; the circuit depth, ansatz design, and parameterization affect expressivity and trainability.

Hybrid training uses classical gradient estimation (finite differences or parameter-shift rules).

3.2.3 Quantum annealing and QUBO encodings

Quantum annealers (e.g., D-Wave) find low-energy states of Ising Hamiltonians, solving Quadratic Unconstrained Binary Optimization (QUBO) problems. Many combinatorial genomics tasks can be cast as QUBOs read ordering, optimal primer selection, small-scale assembly subproblems and quantum annealers have been applied to classification and combinatorial instances in bioinformatics.

3.2.4 Sampling and amplitude estimation subroutines

Quantum amplitude estimation and sampling may in principle accelerate Monte Carlo–type computations, relevant to Bayesian posterior estimation in population genetics. However, practical use requires careful mapping and often fault-tolerant hardware.

4. Mapping Genomic Tasks to Quantum and Hybrid Formulations

We now describe concrete problem encodings and hybrid algorithmic blueprints for core genomic tasks.

4.1 Sequence comparison and similarity search

Problem. Given a query sequence and a database of sequences (or k-mers), find nearest neighbors or matches.

Classical bottleneck. All-pairs comparisons scale as O(NM)O(NM)O(NM) in naive designs; but indexing helps.

Quantum approach. Two promising directions:

1. **Quantum-enhanced kernel classification**: encode k-mer frequency vectors or compressed embeddings into quantum feature maps $|\phi(x)\rangle$ and compute quantum kernel matrices used by kernel SVMs or kernel PCA. For small-sample, high-dimensional settings (e.g., rare variant classification), quantum kernels can provide richer separations.
2. **Grover-style amplitude amplification for search**: theoretical quadratic speedup in unstructured search; however, practical speedup for genomics databases is limited by data-loading (state preparation) costs and oracle construction overheads. Realistic assessments show that Grover's algorithm rarely provides end-to-end gains unless data can be prepared quantum-efficiently.

Practical hybrid pattern. Use classical preprocessing (hashing, locality-sensitive hashing) to reduce candidate sets, then apply quantum kernel methods to classify or refine matches on the reduced set.

4.2 De novo assembly and read ordering

Problem. Reconstruct genome from reads by finding an ordering/path that maximizes overlap consistency.

Classical formulation. Graph assembly problems reduce to Hamiltonian path / TSP-like formulations over overlap graphs.

Quantum formulation. Map the assembly subproblem to a QUBO instance: nodes represent reads or contigs; binary variables encode ordering or selection; overlap scores become edge weights; constraints (no cycles, coverage) are encoded as penalty terms. Quantum annealers (or QAOA on gate-based devices) can then search for low-energy assignments that correspond to plausible assemblies. Industry white-papers and proof-of-concepts have explored TSP-style encodings for read ordering and assembly optimizations.

Practical caveats. Read counts in modern datasets are enormous (millions), so direct QUBO on entire assemblies is infeasible. Decomposition strategies divide and conquer (local assembly regions), hierarchical scaffolding, and classical pre-filtering are necessary to make quantum subproblems tractable.

4.3 Variant calling and genotyping

Problem. From read pileups, infer genotype probabilities and call variants.

Potential quantum uses.

- **Quantum-accelerated probabilistic inference.** Variational QML architectures may assist in learning complex likelihood surrogates (e.g., modeling indel error distributions) in small-data regimes.
- **Quantum kernels for variant classification.** Use QML to classify candidate variant calls (true variant vs sequencing artifact) in difficult contexts (low allele fraction, noisy long reads) where classical features fail.

Recent works demonstrate that QML can provide performance improvements on classification tasks in biology when sample size is small and feature dimension is large common in rare variant contexts. However, rigorous benchmarking against classical ensembles is required.

4.4 Haplotype phasing and combinatorial assembly

Haplotype phasing (reconstructing maternal/paternal haplotypes) reduces to combinatorial optimization (minimum error correction). QUBO encodings and annealing have been proposed as potential accelerators for moderately sized phasing blocks. Hybrid strategies delegate local phasing blocks to quantum annealers while classical methods handle long-range scaffolding.

4.5 Population genetics and clustering

Quantum clustering algorithms and QML classifiers may be applied to detect population structure, admixture, or subtle differentiation. The promise is that quantum kernels might reveal structure masked in classical feature spaces, aiding detection of cryptic population substructure that impacts disease association studies. However, recent reviews caution that claims of advantage are sensitive to encoding choices and data preprocessing; theoretical speedups must be weighed against practical overheads.

4.6 Multi-omics integration and representation learning

Integrating genomics with transcriptomics, epigenomics, and proteomics is central to precision medicine. Variational quantum circuits can be used as feature extractors (quantum encoders) in hybrid autoencoder frameworks to learn compact cross-modal representations. These representation learning strategies can feed into downstream tasks such as patient stratification or drug response prediction.

5. Hybrid AI Architectures and Algorithmic Blueprints

Given current hardware realities, *hybrid* quantum–classical architectures are the pragmatic path. Below we give concrete blueprints, training strategies, and engineering patterns.

5.1 Where to place quantum modules in classical pipelines

Identify *compute-intensive* or *combinatorially hard* subroutines:

- **Discrete combinatorial kernels:** assembly subproblems, primer design, haplotype block phasing → QUBO / annealing.
- **Kernel matrix computations and feature maps:** small-sample high-dimensional classification tasks → quantum kernel evaluations.
- **Sampling-heavy inference:** where amplitude estimation may eventually speed Monte Carlo, but current NISQ hardware is limited.

The architecture follows a **classical orchestration layer** that performs data ingestion, preprocessing, and orchestration; a **quantum execution layer** for targeted quantum subroutines; and a **postprocessing layer** for classical refinement and aggregation.

5.2 Two canonical hybrid patterns

5.2.1 Quantum-assisted classifier (QAC)

1. Preprocess genomic data to produce feature vectors xxx (k-mer counts, embeddings).
2. Compute quantum kernel entries K(xi,xj)K(x_i,x_j)K(xi,xj) on a QPU using a chosen feature map circuit.
3. Train a classical kernel SVM or kernel ridge regressor using the quantum kernel.
4. Optionally retrain or calibrate with classical cross-validation.

This pattern is effective when training dataset size nnn is moderate (hundreds–thousands) and features are high-dimensional.

5.2.2 Quantum-accelerated combinatorial optimizer (QACO)

1. Formulate the discrete subproblem as QUBO: minimize xTQx+qTxx^\top Q x + q^\top xxTQx+qTx subject to constraints encoded via penalties.
2. Use quantum annealer or QAOA to sample low-energy solutions.
3. Validate and refine solutions classically (local search, greedy heuristics).
4. Integrate into larger pipeline (e.g., assembly refinement).

This pattern is applicable to constrained combinatorial kernels embedded within classical workflows.

5.3 Training and optimization strategies

- **Batched hybrid training**: avoid calling the QPU in every gradient step; cache kernel matrices or use mini-batch estimators to reduce quantum calls.
- **Surrogate differentiable approximations**: where end-to-end differentiability is desired, replace quantum subroutines with classical differentiable surrogates during gradient updates and swap in quantum modules for evaluation.
- **Ensemble and consensus**: aggregate multiple quantum solver runs (to smooth stochasticity) and combine with classical heuristics for robust solutions.

5.4 Hardware and software ecosystem

Leverage robust SDKs and hardware stacks (Qiskit, Cirq, Pennylane) and classical accelerators (GPUs/TPUs). Match problem size to device topology (qubit count, connectivity) and plan embeddings (minor-embedding on annealers, logical-to-physical qubit mapping) carefully to minimize chain breaks and penalty tuning overhead.

6. Experimental Protocols, Benchmarks, and Evaluation Metrics

Rigorous evaluation is critical to avoid spurious claims of quantum advantage. We propose best practices.

6.1 Problem instance design and dataset curation

- **Taxonomy of instances**: small-sample high-dimension vs large-sample low-dimension; combinatorial block sizes for assembly subproblems.
- **Public and synthetic datasets**: use publicly available genomic datasets (1000 Genomes, ENA, NIH SRA) plus synthetic instances with ground truth to test assembly and phasing.
- **Preprocessing transparency**: document encoding choices (angle vs amplitude encoding, k-mer size), normalization, and dimensionality reduction steps.

6.2 Baselines and controls

- Compare QML methods against strong classical baselines: kernel SVM with classical kernels (RBF, polynomial), gradient-boosted trees, convolutional/recurrent neural networks, and specialized genomics tools (BWA, GATK, SPAdes).

- Use classical acceleration techniques (approximate kernels, random Fourier features) as baselines for fair assessment.

6.3 Evaluation metrics

- **Predictive tasks**: accuracy, AUC, precision/recall, calibration.
- **Combinatorial tasks**: objective value gap, solution feasibility, running time (wall-clock, including embedding overhead), and energy/stability.
- **Robustness**: sensitivity to noise, error models (simulate sequencing error profiles).
- **Resource accounting**: total QPU time, number of calls, classical compute time; include data-loading costs.

6.4 Statistical testing and reproducibility

- Use repeated randomized experiments, bootstrapping, and non-parametric tests to evaluate significance.
- Publish code, parameter files, and raw instance definitions (where privacy allows) to enable replication.
- When using cloud quantum services, log device calibration metadata (dates, qubit quality) to contextualize results.

6.5 Interpreting performance

Even if raw wall-clock time is larger on QPU solutions, identify regimes where *quality per unit resource* or *best-in-class solution quality* is improved (e.g., combinatorial solution quality under tight time budgets for certain instances). Carefully avoid overstating asymptotic or general advantage.

7. Practical Limitations, Security, and Ethical Considerations

7.1 NISQ hardware limitations and noise

NISQ devices are noisy: gate infidelity and decoherence limit circuit depth and practical problem sizes. This constrains VQC expressivity and increases circuit sampling requirements for accurate kernel estimation. Error mitigation strategies (zero-noise extrapolation, readout error correction) are important but add overhead. Nature

7.2 Data encoding overheads

Encoding classical genomic data into quantum states can be expensive (amplitude encoding requires normalization and multi-qubit gates). For many genomics datasets, the cost of state preparation outweighs benefits unless encoding can be performed efficiently or the quantum module operates on a compressed representation (e.g., sparse k-mer embeddings).

7.3 Security, privacy, and cloud/native concerns

Genomic data are highly sensitive. Quantum cloud services raise privacy questions: data may be uploaded to remote QPU backends. Homomorphic encryption and secure multi-party computation remain immature for quantum pipelines. Practitioners should apply privacy-by-design and possibly federated patterns to avoid raw data transfer. The cloud-native security perspective is critical: orchestration, provenance, and logging must satisfy regulatory requirements. (See Samuel, 2021; Samuel, 2022 on cloud and secure AI patterns.) PMC+1

7.4 Algorithmic fairness and biological bias

Training data for genomics often reflect population biases (e.g., European ancestry overrepresentation). QML or hybrid systems trained on biased cohorts risk propagating or amplifying inequities in precision medicine. Careful dataset curation, fairness audits, and subgroup analyses are essential.

7.5 Energy consumption and environmental considerations

Quantum hardware consumes cryogenic cooling and specialized infrastructure; the energy-cost tradeoffs

compared to large-scale classical compute must be considered in appraisal of sustainability.

8. Roadmap and Recommendations for the Field

Given both promise and practical constraints, we propose the following roadmap for responsible research and translation.

1. **Focus on small-to-moderate instance regimes where quantum feature maps or annealing can add value** (e.g., rare variant classification, local assembly blocks). Benchmark rigorously against tuned classical baselines.
2. **Invest in encoding research**: design biologically meaningful embeddings (k-mer hashing, graph embeddings) that map efficiently into quantum circuits or compressed quantum states.
3. **Adopt hybrid engineering patterns**: keep quantum modules modular and replaceable; use classical surrogates for training where required; minimize QPU calls via caching and surrogate approximations.
4. **Promote open benchmarking datasets and instance repositories** for quantum genomics (synthetic ground-truth instances and privacy-preserving real data variants).
5. **Advance privacy and security practices**: integrate federated learning, secure enclaves, and explicit governance frameworks before sensitive genomic data is processed on remote QPUs. (See Samuel 2021/2022 for cloud security perspectives.)
6. **Interdisciplinary collaborations**: bring together quantum algorithm designers, computational biologists, clinicians, and ethicists to co-design problem formulations that are meaningful biologically and feasible technically.
7. **Transparent reporting**: publish device metadata, embedding details, and all preprocessing to ensure reproducibility and careful interpretation of any claimed quantum benefits.

9. Conclusion

Quantum machine learning and hybrid quantum–classical AI systems introduce interesting new primitives quantum feature maps, variational ansätze, and quantum annealing solvers that could assist specific subproblems in genomic data analysis. At present, practical contributions will most likely be *incremental and specialized*, adding value in small-sample, high-dimensional settings or where discrete combinatorial cores exist and can be decomposed into tractable QUBO instances. True transformative advantages across genomics will depend on advances in error-corrected quantum hardware, improved encoding strategies, and careful hybrid engineering.

This manuscript has articulated mappings from genomics problems to quantum formulations, proposed hybrid architecture and engineering patterns, and emphasized rigorous experimental protocols and ethical guardrails. We recommend that the genomics and quantum communities pursue focused, reproducible pilot studies on well-scoped tasks (local assembly, rare variant classification, phasing blocks) while advancing cross-disciplinary collaboration to ensure results are biologically meaningful, secure, and equitable.

References

1. Havlíček, V., Córcoles, A. D., Temme, K., Harrow, A. W., Kandala, A., Chow, J. M., & Gambetta, J. M. (2019). Supervised learning with quantum-enhanced feature spaces. *Nature*, 567, 209–212. https://www.nature.com/articles/s41586-019-0980-2
2. Havlíček, V. et al., *arXiv*: "Supervised learning with quantum-enhanced feature spaces." (2018/2019). https://arxiv.org/abs/1804.11326
3. Fatunmbi, T. O. (2025). Quantum computing and Artificial Intelligence: Toward a new computational paradigm. *World Journal of Advanced Research and Reviews, 27*(01), 687–695. https://doi.org/10.30574/wjarr.2025.27.1.2498
4. Fatunmbi, T. O. (2022). Impact of data science and cybersecurity in e-commerce using machine learning techniques. *World Journal of Advanced Research and Reviews, 13*(1), 832–846. https://doi.org/10.30574/wjarr.2022.13.1.0607
5. Gupta, R. S., et al. (2025). A systematic review of quantum machine learning for clinical applications. *npj Digital Medicine* (2025). https://www.nature.com/articles/s41746-025-01597-z
6. Li, R. Y., et al. (2021). Quantum processor-inspired machine learning in the high-energy physics and biology domains. *Communications Physics / PMC* (example experiments with D-Wave formulations and small datasets). https://pmc.ncbi.nlm.nih.gov/articles/PMC8212142/
7. Nałęcz-Charkiewicz, K., et al. (2024). Quantum computing in bioinformatics: a systematic review. *Briefings in Bioinformatics / BIB* (mapping QC algorithms to bioinformatics tasks). https://academic.oup.com/bib/article/25/5/bbae391/7733456
8. Maurizio, A., & Mazzola, G. (2025). Quantum computing for genomics: conceptual challenges and practical perspectives. *arXiv/ResearchGate preprint* (conceptual survey). https://arxiv.org/html/2507.04111v1
9. Jha, R. K., et al. (2024). Quantum-enhanced feature maps for improved quantum kernels and classifiers. *Quantum Machine Learning* (proof-of-concept feature maps and analysis). https://pure.ulster.ac.uk/en/publications/quantum-enhanced-feature-maps-for-improved-quantum-kernels-and-ad
10. Industry/white-paper references on quantum assembly optimization: NTT DATA white paper (Quantum computing PoC for genome assembly). https://www.nttdata.com/global/en/-/media/nttdataglobal/1_files/industries/life-science/ls-white-paper-2023-1_quantum-computing-poc_italy.pdf
11. D-Wave case studies and QUBO approaches to biological classification (conference materials). https://www.dwavequantum.com/media/elvpkftl/16_wed_am_usc.pdf
12. Salloum, H. (2024). Quantum annealing in machine learning: QBoost and classification experiments. *ScienceDirect / Procedia* (example QA experiments). https://www.sciencedirect.com/science/article/pii/S1877050924023330
13. Recent arXiv empirical study: Singh, N., Pokhrel, S. R. (2025). Modeling Quantum Machine Learning for Genomic Data Analysis (Qiskit implementations and mapping experiments). https://arxiv.org/pdf/2501.08193
14. Samuel, A. J. (2021). Cloud-Native AI solutions for predictive maintenance in the energy sector: A security perspective. *World Journal of Advanced Research and Reviews, 9*(03), 409–428. https://doi.org/10.30574/wjarr.2021.9.3.0052

15. Samuel, A. J. (2022). AI and machine learning for secure data exchange in decentralized energy markets on the cloud. *World Journal of Advanced Research and Reviews, 16*(2), 1269–1287. https://doi.org/10.30574/wjarr.2022.16.2.1282

16. Frontiers piece: "The promise of quantum computing for population genetics" (2025) conceptual article exploring QML for population genetics structure detection.