



Patent Claims

Independent Claims

1. System Claim (Overall Architecture)

A computational virology system comprising:

- a secure web-based interface configured for receiving viral genomic sequences, protein structural data, and clinical datasets from authenticated users;
- a cloud-based storage module configured to store genomic, proteomic, pharmacological, and epidemiological data in formats including FASTA, PDB, SDF, CSV, and HL7/FHIR;
- a processing engine implemented in Python and comprising machine learning models, wherein the models are trained to perform mutation prediction, protein folding simulations, molecular docking, and antidote optimization;
- a relational database for metadata management, a graph database for linking mutation–drug–symptom relationships, and an object storage repository for binary biological data;
- a visualization module comprising a three-dimensional molecular viewer, a statistical dashboard, and real-time risk assessment displays;
- wherein the system is configured to predict mutational trajectories of viruses, compute pathogenicity scores, optimize drug candidates, simulate clinical pharmacokinetics, and present results in an interactive, interpretable format.

2. Mutation Prediction Method

A method for predicting viral mutations, comprising:

- ingesting viral genomic data in nucleotide sequence formats including FASTA or GenBank;
- performing sequence alignment using codon substitution matrices selected from PAM, BLOSUM, or custom viral-specific matrices;
- applying Hidden Markov Models (HMMs), Markov Chain Monte Carlo (MCMC) simulations, and Long Short-Term Memory (LSTM) neural networks trained on historical viral lineage datasets;
- forecasting probable single nucleotide polymorphisms (SNPs), insertions, deletions, or frame-shift mutations with probabilistic confidence values;
- outputting predicted mutational hotspots in coding regions associated with receptor-binding domains, conserved motifs, or immune-evasion sites.

3. Protein Folding and Dynamics

A method for protein structure prediction and molecular dynamics simulation, comprising:

- translating predicted nucleotide mutations into corresponding amino acid substitutions;
- generating initial three-dimensional structural models using deep learning-based folding models including AlphaFold or OpenFold;
- simulating conformational dynamics using molecular dynamics frameworks including AMBER, CHARMM, and GROMOS force fields;
- computing metrics including Root Mean Square Deviation (RMSD), Root Mean Square Fluctuation (RMSF), Solvent Accessible Surface Area (SASA), and Gibbs free energy stability (ΔG);
- validating predicted protein structures against known crystal or cryo-EM reference models.

4. Antidote and Drug Design

A method for computational antidote generation and optimization, comprising:

- generating candidate molecules using deep generative models including Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), and Diffusion-based molecular generators;
- encoding molecules as SMILES strings, SDF conformers, or MOL2 structures;
- filtering candidates using cheminformatics constraints including Lipinski's Rule of Five, Veber's criteria, and synthetic accessibility scoring;
- docking drug candidates to viral protein targets using docking algorithms including AutoDock Vina, followed by free-energy rescoring using Molecular Mechanics Poisson-Boltzmann Surface Area (MM-PBSA) and Molecular Mechanics Generalized Born Surface Area (MM-GBSA);
- refining selected candidates using Quantum Mechanics/Molecular Mechanics (QM/MM) hybrid simulations to propose chemical modifications such as hydroxylation, halogenation, or methylation to improve drug bioavailability, solubility, and reduced off-target toxicity.

5. In-Silico Clinical Simulation

A method for simulating clinical pharmacological behavior of drug candidates, comprising:

- constructing Physiologically Based Pharmacokinetic (PBPK) and Pharmacodynamic (PK/PD) compartmental models of human tissues and organs including liver, kidney, lung, and brain;
- simulating absorption, distribution, metabolism, and excretion (ADME) pathways using differential equation solvers;

- modeling immunogenicity by agent-based simulation of cytokine signaling pathways;
- predicting dose–response curves, therapeutic indices, and toxicological endpoints;
- generating adverse effect predictions prior to in-vivo or human clinical trials.

6. Risk Assessment and Epidemiological Forecasting

A method for computing viral pathogenicity and outbreak risk, comprising:

- deriving a Deadliness Score by aggregating parameters including receptor-binding affinity (K_d), viral replication rate (R_0), cytopathic effect (CPE) indices, and immune-escape probabilities;
- integrating said Deadliness Score into stochastic SEIR (Susceptible–Exposed–Infectious–Recovered) epidemiological models;
- dynamically adjusting transmission coefficients based on predicted mutational impact;
- generating outbreak forecasts including temporal incidence curves, geographic risk heatmaps, and resource allocation recommendations for healthcare systems.

7. Visualization System

A system for interactive visualization of computational virology results, comprising:

- a WebGL-based molecular viewer configured to render protein structures, ligand binding poses, and conformational changes in real time;
- a statistical dashboard configured to display mutational probabilities, drug candidate affinities, and epidemiological forecast graphs;
- a risk alert interface configured to highlight high-risk mutations and drug candidates using visual encoding such as color intensity, heatmaps, or scoring scales;
- wherein said visualization is dynamically linked to the processing engine and databases to ensure real-time interpretability of computational predictions.

8. Continuous Learning Feedback Loop

A feedback-driven method wherein validated laboratory assays, clinical trial outcomes, and epidemiological reports are ingested back into the knowledge base, thereby retraining mutation forecasting models, protein folding predictors, and antidote design pipelines using transfer learning, thereby ensuring continuous accuracy improvement over time.

Dependent Claims (Examples)

- 9.** The system of claim 1, wherein the processing engine is implemented in Python using libraries including scikit-learn, PyTorch, TensorFlow, RDKit, and BioPython.
- 10.** The method of claim 2, wherein codon substitution analysis is performed using BLOSUM62 matrices specifically adapted for RNA viruses.
- 11.** The method of claim 3, wherein molecular dynamics simulations are performed with timesteps of 2 femtoseconds and trajectory lengths exceeding 100 nanoseconds.
- 12.** The method of claim 4, wherein docking simulations include flexible side-chain modeling and explicit solvent representation for affinity refinement.
- 13.** The method of claim 5, wherein PBPK models are personalized using population-specific physiological parameters derived from clinical datasets.
- 14.** The method of claim 6, wherein outbreak forecasting integrates real-time mobility and demographic data for improved transmission modeling.
- 15.** The visualization system of claim 7, wherein the molecular viewer supports file formats including PDB, mmCIF, and MOL2.