Open-Source Project

Project Bible

HCLS AI Factory Implementation Reference

Complete architecture, pipeline stages, schemas, scoring formulas, and implementation sequences for building the HCLS AI Factory on NVIDIA DGX Spark — from patient DNA to novel drug candidates.

NVIDIA DGX Spark | Parabricks | BioNeMo

02/2026 | Version 1.0 | Apache 2.0 License

Author: Adam Jones

Table of Contents

1. Project Overview & Goals

What This Platform Does

The HCLS AI Factory is an end-to-end precision medicine platform that takes a patient's raw DNA sequencing data (FASTQ) and produces ranked novel drug candidates — all on a single NVIDIA DGX Spark desktop workstation. Three GPU-accelerated stages execute sequentially: variant calling, RAG-grounded target identification, and generative drug discovery.

Three-Stage Pipeline

Stage	Function	Duration	Key Output
1 — Genomics	BWA-MEM2 alignment + DeepVariant calling	120-240 min	VCF (~11.7M variants)
2 — RAG/Chat	Annotation → Embedding → LLM reasoning	Interactive	Target gene + evidence
3 — Drug Discovery	MolMIM → DiffDock → RDKit scoring	8-16 min	100 ranked drug candidates

End-to-End Flow

```
Patient DNA → Illumina Sequencer → FASTQ (~200 GB)
  → Parabricks fq2bam → BAM
  → DeepVariant → VCF (11.7M variants)
  → ClinVar + AlphaMissense + VEP annotation
  → Milvus vector indexing (3.5M embeddings)
  → Claude RAG reasoning → Target hypothesis
  → RCSB PDB structure retrieval
  → MolMIM molecule generation
  → DiffDock molecular docking
  → RDKit drug-likeness scoring
  → 100 ranked novel drug candidates + PDF report
```

Design Principles

GPU-first: Every compute-intensive step runs on the GB10 GPU

Clinically grounded: ClinVar, AlphaMissense, and VEP provide evidence-based annotation

Reproducible: Nextflow DSL2 orchestration with containerized processes

Open: Apache 2.0 license, open-source tools, public reference databases

Desktop-scale: Runs entirely on a $3,999 DGX Spark

2. DGX Spark Hardware Reference

Specifications

Parameter	Value
CPU	NVIDIA Grace (ARM64 / aarch64), 144 cores
GPU	NVIDIA GB10, 1 GPU
Memory	128 GB unified LPDDR5x (CPU + GPU shared)
Storage	NVMe, high-throughput I/O
Storage Access	GPUDirect Storage (zero-copy GPU access)
Price	$3,999
OS	Ubuntu-based (NVIDIA DGX OS)

Critical: ARM64 Architecture

ALL containers must be ARM64-compatible. The Grace CPU is aarch64, not x86_64. This affects base Docker images, Python wheels, NVIDIA container images (NGC ARM64 variants), and any compiled C/C++ extensions.

Unified Memory Model

The 128 GB LPDDR5x is shared between CPU and GPU — there is no separate GPU VRAM. No explicit CPU→GPU data transfers needed for many operations. Memory pressure from CPU workloads reduces GPU-available memory. Parabricks fq2bam peaks at ~40 GB, DeepVariant at ~60 GB.

Storage Requirements

Dataset	Size	Notes
GRCh38 reference	3.1 GB	Pre-indexed for BWA-MEM2
FASTQ input (30× WGS)	~200 GB	HG002 paired-end
BAM intermediate	~100 GB	Temporary, deleted after VCF
ClinVar database	~1.2 GB	4.1M clinical variants
AlphaMissense database	~4 GB	71M predictions
Milvus index	~2 GB	3.5M × 384-dim vectors
BioNeMo model cache	~10 GB	MolMIM + DiffDock weights
Total minimum	~320 GB	Plus OS and Docker layers

Deployment Progression

Phase	Hardware	Price	Scope
1 — Proof Build	DGX Spark	$3,999	Single patient, Docker Compose
2 — Departmental	1–2× DGX B200	$500K–$1M	Multiple concurrent patients, Kubernetes
3 — Enterprise	DGX SuperPOD	$7M–$60M+	Thousands concurrent, FLARE federated

3. Repository Layout

```
hcls-ai-factory-public/
├── README.md                          # Project overview
├── LICENSE                            # Apache 2.0
├── docker-compose.yml                 # All services
├── start-services.sh                  # Startup orchestration
├── .env.example                       # Environment template
│
├── hls-orchestrator/                  # Nextflow pipeline
│   ├── main.nf                        # DSL2 entry point
│   ├── nextflow.config                # Profiles and parameters
│   ├── run_pipeline.py                # Python CLI launcher
│   └── modules/                       # genomics/rag_chat/drug_discovery/reporting
│
├── genomics-pipeline/                 # Stage 1: Parabricks
│   ├── src/run_parabricks.py          # fq2bam + DeepVariant
│   └── src/web_portal.py              # Flask portal (:5000)
│
├── rag-chat-pipeline/                 # Stage 2: RAG + Claude
│   ├── src/rag_engine.py              # Core RAG (23 KB)
│   ├── src/milvus_client.py           # Vector DB client (13 KB)
│   ├── src/annotator.py               # ClinVar+AM+VEP (23 KB)
│   ├── src/knowledge.py               # 201 genes (88 KB)
│   └── src/streamlit_chat.py          # Chat UI (:8501)
│
├── drug-discovery-pipeline/           # Stage 3: BioNeMo + RDKit
│   ├── src/pipeline.py                # 10-stage orchestration (18 KB)
│   ├── src/nim_clients.py             # MolMIM+DiffDock clients (15 KB)
│   ├── src/molecule_generator.py      # SMILES generation (11 KB)
│   ├── src/cryoem_evidence.py         # Cryo-EM scoring (6 KB)
│   └── src/models.py                  # Pydantic models (8 KB)
│
├── landing-page/                      # Entry point (:8080)
├── monitoring/                        # Prometheus + Grafana
└── docs/                              # Documentation (122 KB+)
```

4. Docker Compose Services

Port Allocation

Service	Port	Protocol	Stage
Landing Page	8080	HTTP (Flask)	Orchestration
Genomics Portal	5000	HTTP (Flask)	Stage 1
RAG REST API	5001	HTTP REST	Stage 2
Milvus Vector DB	19530	gRPC	Stage 2
Attu (Milvus UI)	8000	HTTP	Stage 2
Streamlit Chat	8501	HTTP	Stage 2
MolMIM NIM	8001	HTTP REST	Stage 3
DiffDock NIM	8002	HTTP REST	Stage 3
Discovery UI	8505	HTTP (Streamlit)	Stage 3
Discovery Portal	8510	HTTP	Stage 3
Grafana	3000	HTTP	Monitoring
Prometheus	9099	HTTP	Monitoring
Node Exporter	9100	HTTP	Monitoring
DCGM Exporter	9400	HTTP	Monitoring

Key Container Images

Service	Image	Notes
Parabricks	nvcr.io/nvidia/clara/clara-parabricks:4.6.0-1	GPU-accelerated genomics
Milvus	milvusdb/milvus:v2.4-latest	Vector database
MolMIM	nvcr.io/nvidia/clara/bionemo-molmim:1.0	Molecule generation NIM
DiffDock	nvcr.io/nvidia/clara/diffdock:1.0	Molecular docking NIM
Grafana	grafana/grafana:10.2.2	Dashboards
Prometheus	prom/prometheus:v2.48.0	Metrics TSDB

Service Startup Order

The start-services.sh script orchestrates startup in dependency order:

1. Infrastructure (Milvus, monitoring)
2. Stage 1 services (Parabricks, genomics portal)
3. Stage 2 services (RAG engine, Streamlit chat)

4. Stage 3 services (BioNeMo NIMs, discovery UI)
5. Landing page (health monitor for all 10 services)

Health Monitoring

The landing page at port 8080 monitors 10 services:

Service	Health Endpoint	Interval
Parabricks	Port 5000 /health	30s
Milvus	Port 19530 gRPC ping	30s
RAG API	Port 5001 /health	30s
Chat UI	Port 8501 /healthz	30s
MolMIM NIM	Port 8001 /v1/health/ready	30s
DiffDock NIM	Port 8002 /v1/health/ready	30s
Discovery UI	Port 8505 /healthz	30s
Grafana	Port 3000 /api/health	30s
Prometheus	Port 9099 /-/healthy	30s
DCGM Exporter	Port 9400 /metrics	30s

5. Stage 1: Genomics Pipeline

Overview

Stage 1 takes raw FASTQ files from a sequencer and produces a Variant Call Format (VCF) file using NVIDIA Parabricks — a GPU-accelerated implementation of industry-standard bioinformatics tools.

Input Specifications

Parameter	Value
Sample	HG002 (GIAB reference standard)
Coverage	30× whole-genome sequencing (WGS)
Read Length	2×250 bp paired-end
File Size	~200 GB (FASTQ pair)
Reference Genome	GRCh38 (3.1 GB, pre-indexed)
Format	FASTQ (gzip-compressed)

Step 1: BWA-MEM2 Alignment (fq2bam)

bash

```
pbrun fq2bam \
  --ref /reference/GRCh38.fa \
  --in-fq /data/HG002_R1.fastq.gz /data/HG002_R2.fastq.gz \
  --out-bam /output/HG002.bam \
  --num-gpus 1
```

Metric	Value
Duration	20-45 minutes
GPU Utilization	70-90%
Peak Memory	~40 GB
Output	Sorted BAM + BAI index
Algorithm	BWA-MEM2 (GPU-accelerated)

Step 2: DeepVariant Variant Calling

bash

```
pbrun deepvariant \
  --ref /reference/GRCh38.fa \
  --in-bam /output/HG002.bam \
  --out-variants /output/HG002.vcf.gz \
  --num-gpus 1
```

Metric	Value
Duration	10-35 minutes
GPU Utilization	80-95%
Peak Memory	~60 GB
Output	VCF (gzip-compressed + tabix index)
Algorithm	Google DeepVariant (CNN-based, >99% accuracy)

VCF Output Statistics

Metric	Count
Total Variants	~11.7M
High-Quality (QUAL>30)	~3.5M
SNPs	~4.2M
Indels	~1.0M
Coding Region Variants	~35,000
Multi-allelic Sites	~150,000

Parabricks Container

Image: nvcr.io/nvidia/clara/clara-parabricks:4.6.0-1

GPU: Required (CUDA). Volumes: /reference, /data, /output. Port: 5000 (Flask web portal).

6. Stage 2: RAG/Chat Pipeline

Overview

Stage 2 annotates VCF variants with clinical and functional databases, indexes them in a Milvus vector database, and uses Anthropic Claude with RAG to identify druggable gene targets supported by evidence.

Architecture

```
VCF (11.7M variants)
  → Quality filter (QUAL>30) → 3.5M variants
  → ClinVar annotation → clinical significance
  → AlphaMissense annotation → pathogenicity prediction
  → VEP annotation → functional consequences
  → BGE-small-en-v1.5 embedding → 384-dim vectors
  → Milvus IVF_FLAT indexing → 3.5M searchable embeddings
  → Claude RAG query → target hypothesis with evidence chain
```

Annotation Funnel

Stage	Variant Count	Filter
Raw VCF	~11.7M	—
Quality filter	~3.5M	QUAL > 30
ClinVar match	~35,616	Clinical significance annotated
AlphaMissense match	~6,831	AI pathogenicity predicted
Coding + pathogenic	~2,400	Actionable subset

Embedding Model

Parameter	Value
Model	BGE-small-en-v1.5
Dimensions	384
Index Type	IVF_FLAT
Index Params	nlist=1024
Search Params	nprobe=16

Distance Metric	COSINE
Total Embeddings	~3.5M

Query Flow

1. User asks a natural language question in the Streamlit chat
2. Query is expanded using 10 therapeutic area keyword maps
3. BGE-small-en-v1.5 embeds the expanded query
4. Milvus performs approximate nearest-neighbor search (top_k=20)
5. Retrieved variant contexts are assembled into a RAG prompt
6. Claude processes the prompt with knowledge base grounding
7. Response includes gene target, evidence chain, and confidence

7. Milvus Vector Database Schema

Collection: genomic_evidence

17 fields capturing genomic position, annotation, and embedding:

Field	Type	Description
id	INT64 (PK, auto)	Primary key
embedding	FLOAT_VECTOR(384)	BGE-small-en-v1.5 embedding
chrom	VARCHAR(10)	Chromosome (chr1-22, chrX, chrY)
pos	INT64	Genomic position
ref	VARCHAR(1000)	Reference allele
alt	VARCHAR(1000)	Alternate allele
qual	FLOAT	Variant quality score
gene	VARCHAR(100)	Gene symbol
consequence	VARCHAR(200)	Functional consequence
impact	VARCHAR(20)	HIGH, MODERATE, LOW, MODIFIER
genotype	VARCHAR(10)	Sample genotype (0/1, 1/1)
text_summary	VARCHAR(2000)	Human-readable description
clinical_significance	VARCHAR(200)	ClinVar classification
rsid	VARCHAR(20)	dbSNP identifier
disease_associations	VARCHAR(2000)	Associated diseases
am_pathogenicity	FLOAT	AlphaMissense score (0-1)
am_class	VARCHAR(20)	pathogenic/ambiguous/benign

Index Configuration

```python
index_params = {
    "index_type": "IVF_FLAT",
    "metric_type": "COSINE",
    "params": {"nlist": 1024}
}

search_params = {
    "metric_type": "COSINE",
    "params": {"nprobe": 16}
}
```

Milvus Infrastructure

Component	Port	Purpose
Milvus standalone	19530	gRPC vector operations
Attu UI	8000	Web-based Milvus management
etcd	2379	Metadata storage
MinIO	9000	Object storage for indexes

8. Variant Annotation Pipeline

ClinVar Integration

Parameter	Value
Database	ClinVar (NCBI)
Total Variants	4.1M clinical variants
Match Rate	~35,616 / 3.5M variants (1.0%)
Classifications	Pathogenic, Likely pathogenic, VUS, Likely benign, Benign
Update Frequency	Monthly releases

AlphaMissense Integration

Parameter	Value
Database	AlphaMissense (DeepMind)
Total Predictions	71,697,560 missense variant predictions
Match Rate	~6,831 / 35,616 ClinVar variants (19.2%)
Model	AlphaFold-derived protein structure features
Output	Pathogenicity score (0.0-1.0)

AlphaMissense Thresholds

Class	Score Range	Interpretation
Pathogenic	> 0.564	Likely disease-causing
Ambiguous	0.34 – 0.564	Uncertain significance
Benign	< 0.34	Likely neutral

Ensembl VEP Integration

Parameter	Value
Tool	Ensembl Variant Effect Predictor (VEP)
Purpose	Functional consequence annotation
Impact Levels	HIGH, MODERATE, LOW, MODIFIER
Key Consequences	missense_variant, stop_gained, frameshift_variant, splice_donor_variant

Annotation Pipeline Code Pattern

python

```
    def annotate_variants(vcf_path: str) -> List[AnnotatedVariant]:
        """VCF → ClinVar → AlphaMissense → VEP → Annotated variants"""
        variants = parse_vcf(vcf_path, min_qual=30)        # ~3.5M pass
        variants = annotate_clinvar(variants)                # Clinical significance
        variants = annotate_alphamissense(variants)          # AI pathogenicity
        variants = annotate_vep(variants)                    # Functional consequences
        return variants
```

9. Knowledge Base — 201 Genes, 13 Therapeutic Areas

Gene Distribution

Therapeutic Area	Count	Example Genes
Neurology	36	VCP, APP, PSEN1, MAPT, SOD1, FUS, C9orf72
Oncology	27	EGFR, BRAF, KRAS, TP53, BRCA1, BRCA2, PIK3CA
Metabolic	22	GCK, PPARG, SLC2A2, ABCA1, PCSK9
Infectious Disease	21	ACE2, CCR5, IFITM3, TLR4, TMPRSS2
Respiratory	13	CFTR, SERPINA1, MUC5B, TERT

Rare Disease	12	VCP, HTT, SMN1, DMD, CFTR
Hematology	12	HBB, HBA1, F5, JAK2, CALR
GI/Hepatology	12	HFE, ATP7B, NOD2, SERPINA1
Pharmacogenomics	11	CYP2D6, CYP2C19, CYP3A4, DPYD, TPMT
Ophthalmology	11	RHO, RPE65, RS1, ABCA4
Cardiovascular	10	LDLR, PCSK9, SCN5A, MYBPC3, KCNQ1
Immunology	9	HLA-B, TNF, IL6, JAK1, CTLA4
Dermatology	9	FLG, MC1R, TYR, KRT14

Total: 201 genes, 171 druggable targets (85% druggability rate).

Knowledge Base Entry Structure

python

```
{
    "gene": "VCP",
    "uniprot": "P55072",
    "therapeutic_area": "Neurology",
    "diseases": ["Frontotemporal Dementia", "ALS", "IBMPFD"],
    "druggability": "High",
    "drug_targets": ["D2 ATPase domain", "N-D1 interface"],
    "known_inhibitors": ["CB-5083", "NMS-873"],
    "variant_hotspots": ["R155H", "R191Q", "A232E"],
    "pathway": "Ubiquitin-proteasome system",
    "mechanism": "AAA+ ATPase, protein homeostasis"
}
```

Query Expansion Maps

10 therapeutic area query expansion maps enrich user queries with domain-specific terminology for improved Milvus retrieval.

10. Anthropic Claude LLM Integration

Configuration

Parameter	Value
Model	claude-sonnet-4-20250514
Temperature	0.3
Max Tokens	4096
API	Anthropic Messages API
Role	RAG-grounded clinical reasoning

RAG Prompt Structure

```python
system_prompt = """You are a clinical genomics specialist
analyzing patient variant data. Ground all responses in
the retrieved variant evidence and knowledge base. Cite
specific variants, genes, and clinical classifications.
When recommending drug targets, explain the evidence
chain from variant to disease mechanism to druggability."""

user_prompt = f"""
## Retrieved Variant Evidence (top {top_k} matches)
{formatted_variants}

## Knowledge Base Context
{knowledge_context}

## User Question
{user_question}
"""
```

Response Format

Claude generates structured target hypotheses including gene, confidence level, evidence chain, therapeutic area, diseases, and recommended action for downstream drug discovery.

Note: *Claude is only used in this environment for functional testing. A local LLM that aligns with FDA clinical standards would be used in a clinical setting.*

11. Stage 3: Drug Discovery Pipeline

Overview

Stage 3 takes a target gene hypothesis from Stage 2 and produces 100 ranked novel drug candidates using BioNeMo generative chemistry, molecular docking, and drug-likeness scoring.

10-Stage Pipeline

Stage	Process	Description
1	Initialize	Load target hypothesis, validate inputs
2	Normalize Target	Map gene → UniProt ID → PDB structures
3	Structure Discovery	Query RCSB PDB for Cryo-EM/X-ray structures
4	Structure Preparation	Score and rank structures, select best site
5	Molecule Generation	MolMIM generates novel SMILES from

		seed
6	Chemistry QC	RDKit validates chemical feasibility
7	Conformer Generation	RDKit 3D conformer embedding (ETKDG)
8	Molecular Docking	DiffDock predicts binding poses and affinities
9	Composite Ranking	30% gen + 40% dock + 30% QED weighted scoring
10	Reporting	PDF report generation (ReportLab)

Pipeline Configuration

python

```
PIPELINE_CONFIG = {
    "num_candidates": 100,
    "molmim_endpoint": "http://localhost:8001/v1/generate",
    "diffdock_endpoint": "http://localhost:8002/v1/dock",
    "min_qed": 0.3,
    "min_dock_score": -6.0,         # kcal/mol
    "scoring_weights": {
        "generation": 0.30,
        "docking": 0.40,
        "qed": 0.30
    }
}
```

UniProt Mappings

Gene	UniProt ID	Function
VCP	P55072	AAA+ ATPase, protein homeostasis
EGFR	P00533	Receptor tyrosine kinase
BRAF	P15056	Serine/threonine kinase
KRAS	P01116	GTPase signaling

12. BioNeMo NIM Services

MolMIM (Port 8001) — Molecule Generation

Parameter	Value
Endpoint	POST http://localhost:8001/v1/generate
Model	MolMIM (Molecular Masked Inverse Model)
Input	Seed SMILES string
Output	Novel SMILES candidates

Container	nvcr.io/nvidia/clara/bionemo-molmim:1.0

MolMIM Request/Response

json

```
# Request
{"smiles": "CC(=O)Nc1ccc(O)cc1",
 "num_molecules": 100,
 "temperature": 0.7, "top_k": 50}

# Response
{"molecules": [
  {"smiles": "CC(=O)Nc1ccc(O)c(F)c1", "score": 0.85},
  {"smiles": "CC(=O)Nc1ccc(O)c(Cl)c1", "score": 0.82}
]}
```

DiffDock (Port 8002) — Molecular Docking

Parameter	Value
Endpoint	POST http://localhost:8002/v1/dock
Model	DiffDock (diffusion-based docking)
Input	Ligand SMILES + protein PDB structure
Output	Binding pose + affinity score (kcal/mol)
Container	nvcr.io/nvidia/clara/diffdock:1.0

Docking Score Interpretation

Score (kcal/mol)	Interpretation
-12 to -8	Excellent binding affinity
-8 to -6	Good binding affinity
-6 to -4	Moderate binding affinity
> -4	Weak binding affinity

13. Drug-Likeness Scoring

Lipinski’s Rule of Five

Rule	Threshold	Description
Molecular Weight	≤ 500 Da	Oral absorption limit
LogP	≤ 5	Lipophilicity
H-Bond Donors	≤ 5	NH + OH groups

H-Bond Acceptors	≤ 10	N + O atoms

QED (Quantitative Estimate of Drug-likeness)

Range	Interpretation
> 0.67	Drug-like (favorable properties)
0.49 – 0.67	Moderate drug-likeness
< 0.49	Less drug-like

TPSA (Topological Polar Surface Area)

Range (Å²)	Interpretation
< 140	Good oral bioavailability
60–90	Optimal range
> 140	Poor oral absorption

Composite Scoring Formula

python

```
    def compute_composite_score(gen_score, dock_score, qed_score):
        """30% generation + 40% docking + 30% QED"""
        dock_normalized = max(0.0, min(1.0, (10.0 + dock_score) / 20.0))
        composite = (
            0.30 * gen_score +
            0.40 * dock_normalized +
            0.30 * qed_score
        )
        return composite
```

RDKit Property Calculation

python

```
    from rdkit import Chem
    from rdkit.Chem import Descriptors, QED

    def calculate_properties(smiles: str) -> dict:
        mol = Chem.MolFromSmiles(smiles)
        return {
            "molecular_weight": Descriptors.MolWt(mol),
            "logp": Descriptors.MolLogP(mol),
            "hbd": Descriptors.NumHDonors(mol),
            "hba": Descriptors.NumHAcceptors(mol),
            "tpsa": Descriptors.TPSA(mol),
            "qed": QED.qed(mol),
            "lipinski_pass": all([
                Descriptors.MolWt(mol) <= 500,
                Descriptors.MolLogP(mol) <= 5,
                Descriptors.NumHDonors(mol) <= 5,
                Descriptors.NumHAcceptors(mol) <= 10,
            ])
```

```
}
```

14. Cryo-EM Structure Evidence

Structure Scoring Algorithm

The pipeline automatically retrieves and scores PDB structures:

python

```
    def score_structure(structure: StructureInfo) -> float:
        """Score PDB structure for drug discovery suitability.
        - Resolution: lower is better (max 5 Å cutoff)
        - Inhibitor-bound: +3 bonus
        - Druggable pockets: +0.5 per pocket
        - Cryo-EM method: +0.5"""
        score += max(0, 5.0 - resolution)
        if has_inhibitor_bound: score += 3.0
        score += num_druggable_pockets * 0.5
        if 'Cryo-EM' in method: score += 0.5
        return score
```

VCP Structures (Demo)

PDB ID	Resolution	Method	Description
8OOI	2.9 Å	Cryo-EM	WT VCP hexamer
9DIL	3.2 Å	Cryo-EM	Mutant VCP
7K56	2.5 Å	Cryo-EM	VCP complex
5FTK	2.3 Å	X-ray	VCP + CB-5083 inhibitor

VCP Binding Site

Parameter	Value
Domain	D2 ATPase domain
Mechanism	ATP-competitive inhibition
Pocket Volume	~450 Å³
Druggability Score	0.92
Key Residues	ALA464, GLY479, ASP320, GLY215

15. VCP/FTD Demo Walkthrough

Demo Target: Valosin-Containing Protein (VCP/p97)

Parameter	Value
Gene	VCP
Protein	p97 / Valosin-Containing Protein
UniProt	P55072
Function	AAA+ ATPase, ubiquitin-proteasome pathway
Diseases	Frontotemporal Dementia (FTD), ALS, IBMPFD
Variant	rs188935092 (chr9:35065263 G>A)
ClinVar	Pathogenic
AlphaMissense	0.87 (pathogenic, >0.564 threshold)
Seed Compound	CB-5083 (Phase I clinical VCP inhibitor)

Demo Flow

Stage 1 — Genomics (Demo Mode: ~20 min)

1. Load pre-processed HG002 FASTQ subset
2. Run Parabricks fq2bam alignment
3. Run DeepVariant variant calling
4. Output VCF with ~11.7M variants including rs188935092

Stage 2 — RAG/Chat (Interactive)

1. VCF annotated: ClinVar flags rs188935092 as pathogenic in VCP
2. AlphaMissense scores the missense variant at 0.87 (pathogenic)
3. 3.5M variants embedded and indexed in Milvus
4. User queries: "What are the most promising drug targets?"
5. Claude identifies VCP with full evidence chain
6. Target hypothesis: VCP → FTD → druggable D2 ATPase domain

Stage 3 — Drug Discovery (~10 min)

1. VCP → UniProt P55072 → PDB structure retrieval
2. Cryo-EM structures scored: 8OOI, 9DIL, 7K56, 5FTK
3. 5FTK selected (inhibitor-bound, highest score)
4. CB-5083 seed SMILES → MolMIM generates 100 novel analogs
5. RDKit validates Lipinski, QED, TPSA
6. DiffDock docks each candidate against VCP D2 domain
7. Composite ranking: 30% gen + 40% dock + 30% QED

8. Top candidates: novel VCP inhibitors with improved drug-likeness
9. PDF report generated via ReportLab

Expected Demo Output

```
Pipeline: HCLS AI Factory — VCP/FTD Demo
Target: VCP (P55072) — Frontotemporal Dementia
Seed: CB-5083 (ATP-competitive VCP inhibitor)
Structure: 5FTK (2.3 Å, X-ray, inhibitor-bound)

Results:
- 100 novel VCP inhibitor candidates generated
- 87 pass Lipinski's Rule of Five
- 72 have QED > 0.67 (drug-like)
- Top 10: docking scores -8.2 to -11.4 kcal/mol
- Composite scores range 0.68-0.89
```

16. Pydantic Data Models

Core Models (from models.py)

All data flows use Pydantic models for validation:

TargetHypothesis

python

```python
class TargetHypothesis(BaseModel):
    """Output from Stage 2 — RAG-identified drug target"""
    gene: str                    # e.g., 'VCP'
    uniprot_id: str              # e.g., 'P55072'
    confidence: str              # high, medium, low
    evidence_chain: List[str]
    therapeutic_area: str
    diseases: List[str]
    druggability_score: float    # 0-1 scale
```

RankedCandidate

python

```python
class RankedCandidate(BaseModel):
    """Final ranked drug candidate"""
    rank: int
    smiles: str
    generation_score: float
    dock_score: float            # kcal/mol
    qed: float
    composite_score: float       # 30% gen + 40% dock + 30% QED
    lipinski_pass: bool
    molecular_weight: float
    logp: float
```

PipelineConfig

python

```python
class PipelineConfig(BaseModel):
```

```
    """Pipeline execution configuration"""
    mode: str                    # full, target, drug, demo
    num_candidates: int = 100
    min_qed: float = 0.3
    min_dock_score: float = -6.0
    molmim_url: str = "http://localhost:8001/v1/generate"
    diffdock_url: str = "http://localhost:8002/v1/dock"
```

Additional models: StructureInfo, StructureManifest, MoleculeProperties, GeneratedMolecule, DockingResult, PipelineRun.

17. Nextflow DSL2 Orchestration

Pipeline Modes

Mode	Stages	Description
full	1 → 2 → 3	Complete end-to-end pipeline
target	2 → 3	Skip genomics, use existing VCF
drug	3 only	Skip to drug discovery with known target
demo	1 → 2 → 3	Pre-configured VCP/FTD demonstration
genomics_only	1 only	Run only variant calling

Main Pipeline Entry (main.nf)

groovy

```
#!/usr/bin/env nextflow
nextflow.enable.dsl=2

include { GENOMICS_PIPELINE } from './modules/genomics'
include { RAG_CHAT_PIPELINE } from './modules/rag_chat'
include { DRUG_DISCOVERY_PIPELINE } from './modules/drug_discovery'
include { REPORTING } from './modules/reporting'

workflow {
    if (params.mode in ['full', 'demo', 'genomics_only']) {
        GENOMICS_PIPELINE(params.fastq_r1, params.fastq_r2, params.reference)
    }
    if (params.mode in ['full', 'demo', 'target']) {
        RAG_CHAT_PIPELINE(...)
    }
    if (params.mode in ['full', 'demo', 'target', 'drug']) {
        DRUG_DISCOVERY_PIPELINE(...)
    }
    REPORTING(DRUG_DISCOVERY_PIPELINE.out.candidates)
}
```

Nextflow Profiles

Profile	Description
standard	Default local execution
docker	Docker container execution
singularity	Singularity container execution
dgx_spark	DGX Spark optimized (GPU resources)
slurm	HPC cluster submission
test	Minimal test data

Pipeline Launcher (run_pipeline.py)

bash

```
# Full pipeline
python run_pipeline.py --mode full \
  --fastq-r1 /data/HG002_R1.fastq.gz \
  --fastq-r2 /data/HG002_R2.fastq.gz \
  --reference /reference/GRCh38.fa

# Demo mode (pre-configured VCP/FTD)
python run_pipeline.py --mode demo

# Drug discovery only
python run_pipeline.py --mode drug --target-gene VCP
```

18. Landing Page & Service Health

Landing Page (Port 8080)

The Flask-based landing page serves as the entry point for the HCLS AI Factory, providing a 10-service health status dashboard, pipeline mode selector, quick-start links, real-time status with green/red indicators, and pipeline execution history.

Service Health Check Implementation

python

```
SERVICES = [
    {"name": "Parabricks Portal", "port": 5000},
    {"name": "Milvus Vector DB", "port": 19530},
    {"name": "RAG API", "port": 5001},
    {"name": "Streamlit Chat", "port": 8501},
    {"name": "MolMIM NIM", "port": 8001},
    {"name": "DiffDock NIM", "port": 8002},
    {"name": "Discovery UI", "port": 8505},
    {"name": "Grafana", "port": 3000},
    {"name": "Prometheus", "port": 9099},
    {"name": "DCGM Exporter", "port": 9400},
]
```

19. Monitoring Stack

Grafana (Port 3000)

Parameter	Value
Image	grafana/grafana:10.2.2
Default User	admin / changeme
Dashboards	HCLS AI Factory (GPU, pipeline, services)
Data Source	Prometheus

Prometheus (Port 9099)

Parameter	Value
Image	prom/prometheus:v2.48.0
Internal Port	9090 → External 9099
Retention	30 days
Targets	Node Exporter, DCGM Exporter, service metrics

DCGM Exporter (Port 9400)

Metric	Description
DCGM_FI_DEV_GPU_UTIL	GPU utilization percentage
DCGM_FI_DEV_FB_USED	GPU memory used (bytes)
DCGM_FI_DEV_FB_FREE	GPU memory free (bytes)
DCGM_FI_DEV_GPU_TEMP	GPU temperature (°C)
DCGM_FI_DEV_POWER_USAGE	GPU power draw (watts)
DCGM_FI_DEV_SM_CLOCK	SM clock frequency (MHz)

Key Dashboard Panels

1. GPU Utilization Timeline — fq2bam → DeepVariant → MolMIM/DiffDock bursts
2. Pipeline Stage Progress — Stage 1/2/3 completion with timing
3. Memory Pressure — Unified memory usage across CPU + GPU
4. Service Health Grid — Green/red status for all 10 services
5. Variant Processing Rate — Variants annotated per second
6. Drug Discovery Throughput — Molecules generated/docked per minute

20. Cross-Modal Integration

HCLS AI Factory Ecosystem

The genomics-to-drug-discovery pipeline integrates with the broader HCLS AI Factory, including the Imaging Intelligence Agent:

Cross-Modal Triggers

Trigger	Source	Target	Action
Lung-RADS 4B+	Imaging Agent	Genomics Pipeline	Initiate tumor profiling
Pathogenic Variant	Genomics Pipeline	Drug Discovery	Generate targeted therapies
Drug Candidates	Drug Discovery	Imaging Agent	Combined clinical report

NVIDIA FLARE — Federated Learning

For multi-site deployments (Phase 3), NVIDIA FLARE enables federated model training. Models train locally at each site; only model updates (not patient data) are shared. Raw genomic data never leaves the institution.

21. Configuration Reference

Environment Variables

Variable	Default	Description
ANTHROPIC_API_KEY	(required)	Anthropic API key for Claude
NGC_API_KEY	(required)	NVIDIA NGC key for BioNeMo NIMs
REFERENCE_GENOME	/reference/GRCh38.fa	Path to GRCh38 reference
MILVUS_HOST	localhost	Milvus server hostname
MILVUS_PORT	19530	Milvus gRPC port
MOLMIM_URL	http://localhost:8001	MolMIM NIM endpoint
DIFFDOCK_URL	http://localhost:8002	DiffDock NIM endpoint
CLAUDE_MODEL	claude-sonnet-4-20250514	Claude model identifier
CLAUDE_TEMPERATURE	0.3	LLM temperature
PIPELINE_MODE	full	Pipeline execution mode
NUM_CANDIDATES	100	Drug candidates to generate
MIN_QED	0.3	Minimum QED threshold
MIN_DOCK_SCORE	-6.0	Minimum docking score (kcal/mol)

AlphaMissense Thresholds

```python
AM_PATHOGENIC_THRESHOLD = 0.564
AM_AMBIGUOUS_LOWER = 0.34
AM_AMBIGUOUS_UPPER = 0.564
AM_BENIGN_THRESHOLD = 0.34
```

Scoring Weights

```python
SCORING_WEIGHTS = {
    "generation": 0.30,   # MolMIM generation confidence
    "docking": 0.40,      # DiffDock binding affinity
    "qed": 0.30           # RDKit drug-likeness
}
```

Drug-Likeness Thresholds

```python
LIPINSKI = {"max_mw": 500, "max_logp": 5, "max_hbd": 5, "max_hba": 10}
QED = {"drug_like": 0.67, "moderate": 0.49}
DOCKING = {"excellent": -8.0, "good": -6.0, "moderate": -4.0, "minimum": -6.0}
```

22. Deployment Roadmap

Phase 1: Proof Build

Parameter	Value
Hardware	NVIDIA DGX Spark ($3,999)
Orchestration	Docker Compose
Scale	Single patient, sequential processing
GPU	1× GB10
Memory	128 GB unified

Phase 2: Departmental

Parameter	Value
Hardware	1–2× DGX B200
Orchestration	Kubernetes
Scale	Multiple concurrent patients
GPU	8× B200 per node

Memory	1–2 TB HBM3e

Phase 3: Enterprise / Multi-Site

Parameter	Value
Hardware	DGX SuperPOD
Orchestration	Kubernetes + NVIDIA FLARE
Scale	Thousands of concurrent patients
GPU	Hundreds of B200 GPUs
Privacy	Federated learning (data stays local)

Scaling Considerations

Bottleneck	Phase 1 Solution	Phase 2+ Solution
Genomics throughput	Sequential (1 sample)	Parallel Parabricks instances
Milvus query latency	Single-node Milvus	Cluster with sharding
BioNeMo inference	Single NIM per model	Multiple NIM replicas
Storage I/O	NVMe direct	GPUDirect Storage + RAID

23. Testing Strategy

Unit Tests

Component	Test Focus
VCF Parser	Variant extraction, quality filtering
Annotator	ClinVar/AlphaMissense/VEP lookup accuracy
Milvus Client	Index creation, search recall
MolMIM Client	SMILES generation, request format
DiffDock Client	Docking request/response parsing
RDKit Scoring	Lipinski, QED, TPSA calculations
Composite Scorer	Weight application, normalization

Integration Tests

Test	Validates
VCF → Annotation → Milvus	End-to-end Stage 2 pipeline
Target → PDB → MolMIM → DiffDock	End-to-end Stage 3 pipeline
Health check endpoints	All 10 services responding
Nextflow modes	full, target, drug, demo execution

Demo Mode Validation

The demo pipeline mode uses pre-configured inputs to validate the complete pipeline. Input: HG002 FASTQ subset. Expected: VCP identified as target with rs188935092 evidence. Output: 100 ranked novel VCP inhibitor candidates.

24. Implementation Sequence

Recommended Build Order

1. Infrastructure: Docker Compose, Milvus, monitoring stack
2. Stage 1 — Genomics: Parabricks container, fq2bam, DeepVariant, VCF output
3. Stage 2 — Annotation: ClinVar + AlphaMissense + VEP pipeline
4. Stage 2 — Vector DB: Milvus schema, BGE embedding, IVF_FLAT index
5. Stage 2 — RAG: Claude integration, knowledge base, query expansion
6. Stage 2 — Chat UI: Streamlit interface, REST API
7. Stage 3 — Structure: RCSB PDB retrieval, Cryo-EM scoring
8. Stage 3 — Generation: MolMIM NIM, molecule generation
9. Stage 3 — Docking: DiffDock NIM, binding prediction
10. Stage 3 — Scoring: RDKit properties, composite ranking
11. Stage 3 — Reporting: PDF generation, Discovery UI
12. Orchestration: Nextflow DSL2, pipeline modes, landing page
13. Testing: Unit tests, integration tests, demo mode validation
14. Monitoring: Grafana dashboards, alerting rules

Key Dependencies

```
GRCh38 reference → BWA-MEM2 index → fq2bam alignment
ClinVar + AlphaMissense databases → Annotation pipeline
Milvus running → Embedding indexing → RAG queries
BioNeMo NIMs running → Molecule generation + docking
```

```
All services healthy → Landing page green status
```

This Project Bible is the authoritative technical reference for the HCLS AI Factory. All other documentation assets derive their technical details from this source.