

Designing Better Cancer Vaccines

UCL-CCC-Hackathon-2025
Team TC-AWARE

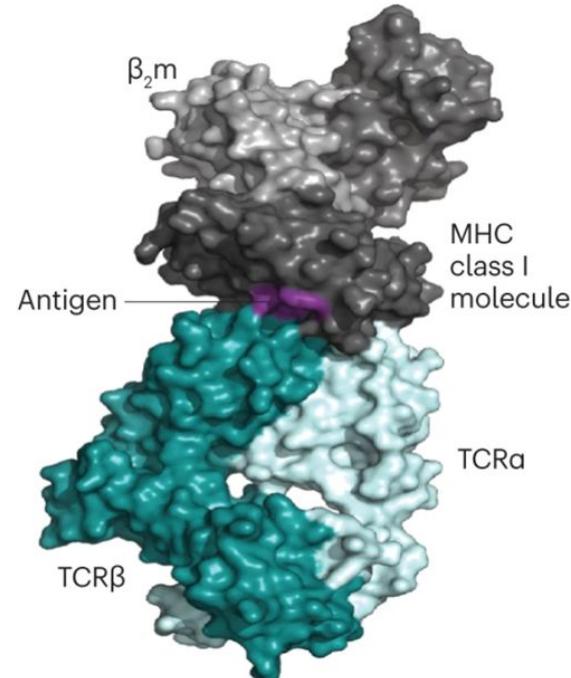
Matthew Cowley, Zehn Wei Yap, Mohammad Alawwami, Zarif Shafiei, Linh Hoang, Julia Sala-Bayo, Graham Bonomo-Jackson, Gleb Gmyzov, Nick Keatley

Problem statement

Cancer vaccines offer an amazing opportunity to develop personalised cures for patients. Our immune system is well equipped to kill cancer but it needs help finding the right target. The most important step is identifying the right target.

- Hard to choose targets as the rough search space of cancer peptides is in excess of 10^{17}

We have formulated an end-to-end pipeline that identifies and ranks potential targets (neoantigens)



Hudson et al. 2023 *Nat. Rev. Imm.*

How our proposal can help

Cancer vaccines can be highly effective, but there is a high rate of non-responders.

We propose to improve vaccine design by using patient immune receptors to select peptides which will drive a strong vaccine response.

Solution could reduce manufacturing time and costs (\$100,000 per patient currently)

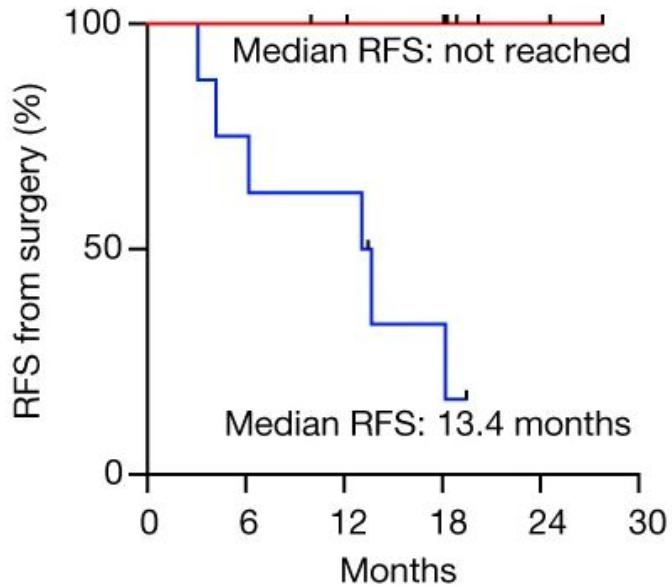
- Enables re-distribution of funds to better improve patient care

Responders ($n = 8$)
Non-responders ($n = 8$)

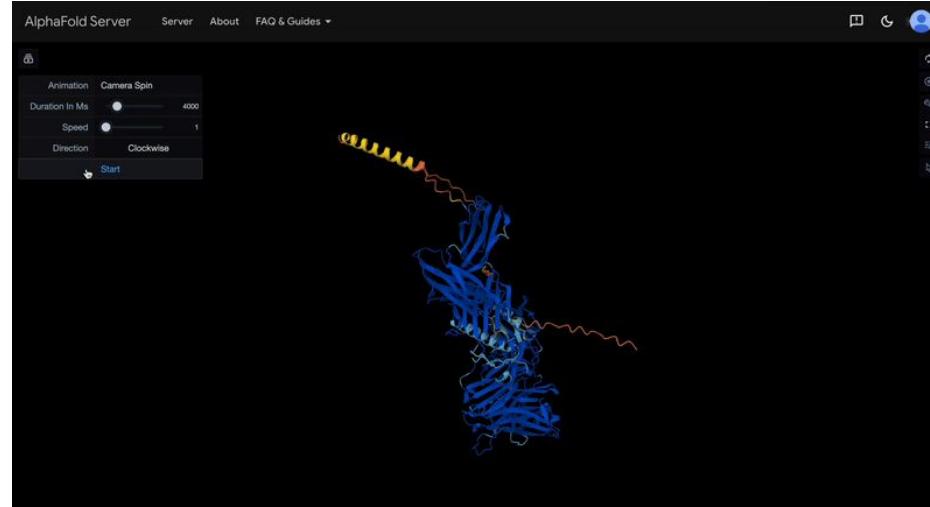
$P = 0.003$

HR: 0.08 (0.01–0.4)

Median follow-up: 18.0 months



Project Overview



Goal	Find new weak spots on cancer cells that the patient's immune system hasn't noticed yet
Output	A prioritised list of cancer targets to put in a vaccine, but specifically excluding targets the immune system has already tried and failed to attack
Key Innovation	Personalised cancer vaccines that target the cancer's fundamental weak spots, specifically choosing targets the patient's immune system hasn't already failed to attack.

Data Requirements

Training Data (model development)

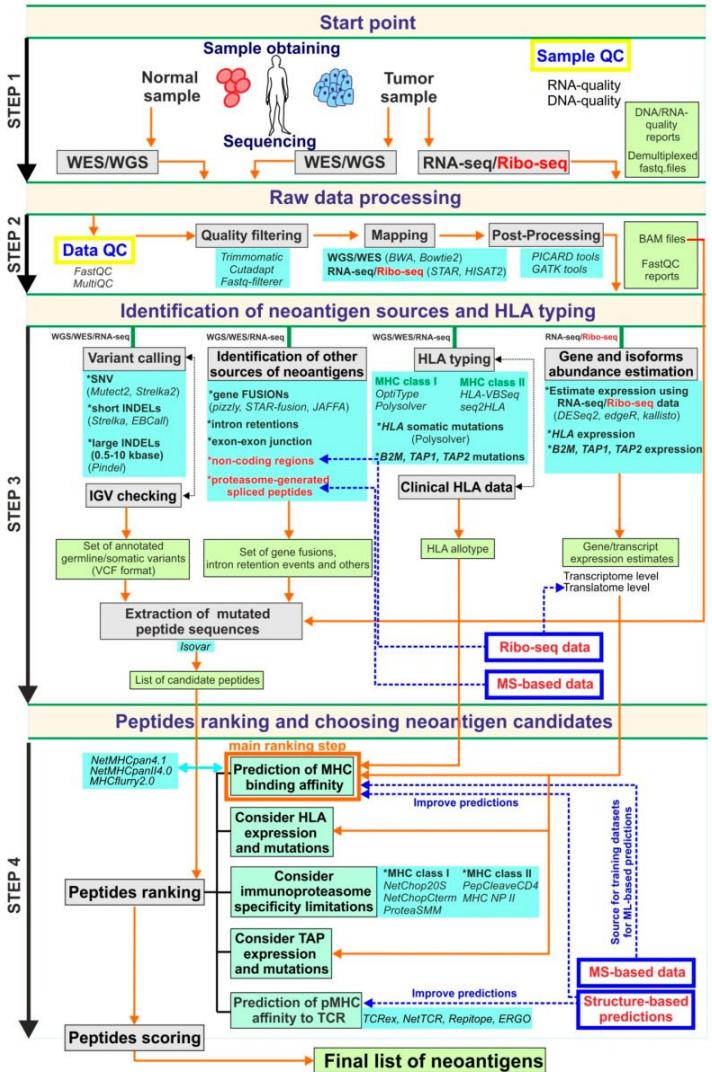
Data Type	Source	Used For
Cancer genomes with validated neoantigens	Cancer genome studies	Neoantigen identification
HLA-peptide binding datasets (multi-individual)	IEDB, published datasets	MHC Find / binding prediction
TCR-peptide binding data	Viral/autoimmune contexts, structural DBs	TCR binding prediction

Data Requirements

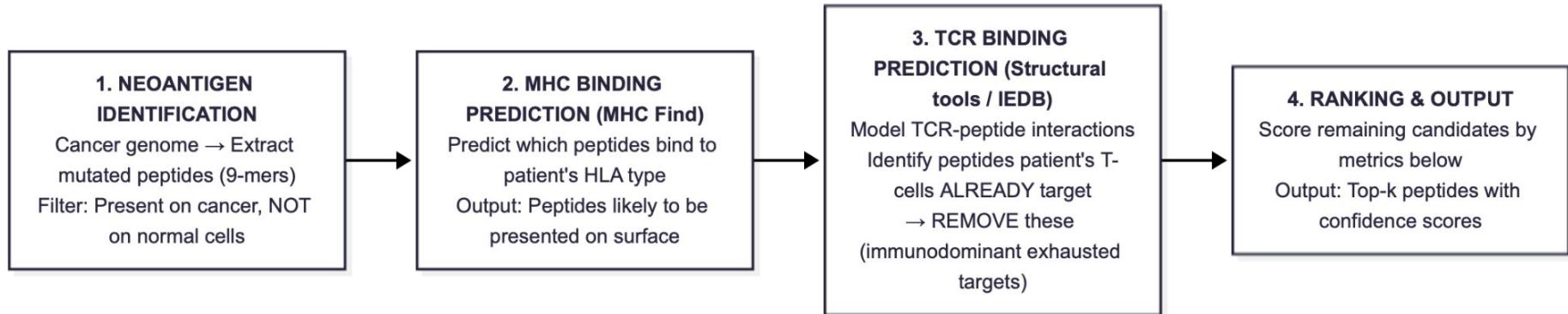
Patient-Specific Data (per-patient input)

Data Type	Source
Cancer genome	Tumour sample (NGS)
TCR repertoire	Blood sample sequencing
HLA/MHC type	Patient genotyping

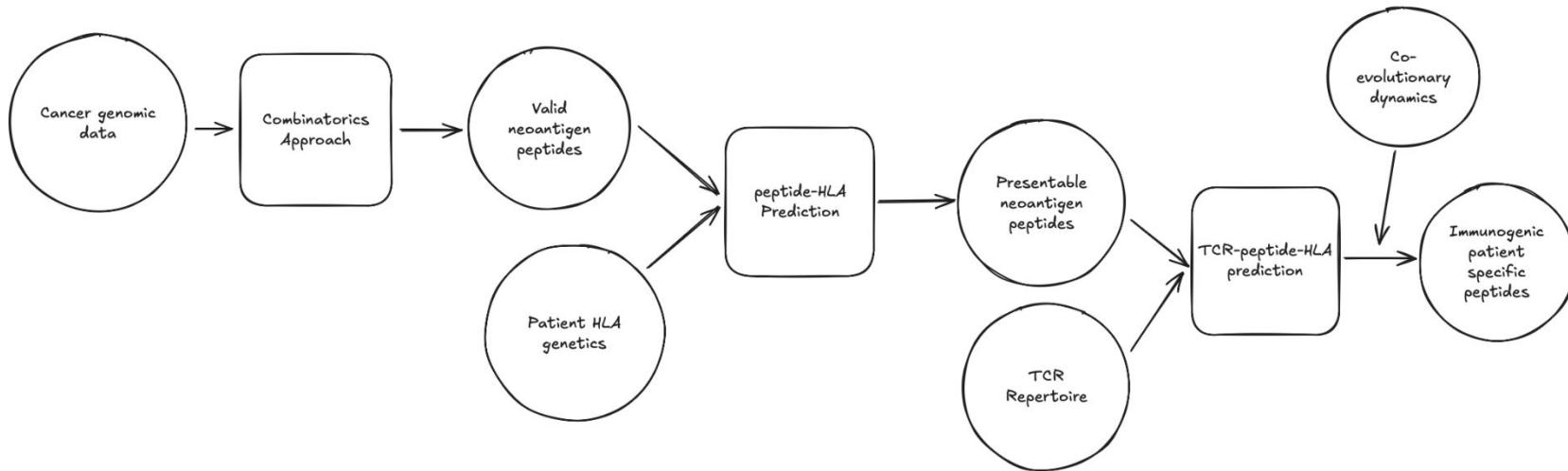
In silico neoantigen prediction



High Level Processing/Development Pipeline



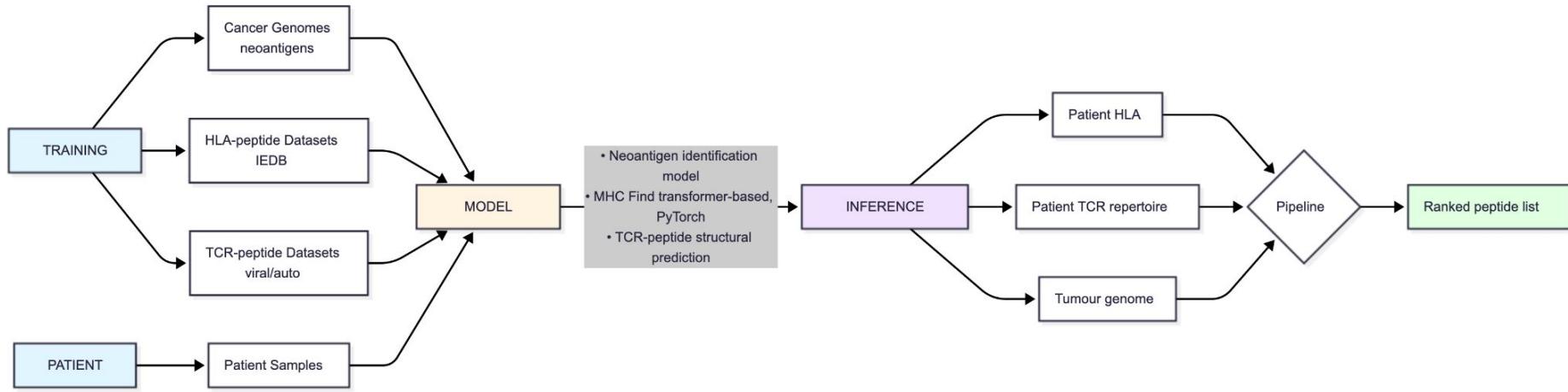
Clinical Development Pipeline



Ranking Metrics

Metric	Description	Why It Matters
MHC-peptide binding affinity	How strongly peptide binds to patient's HLA	Must be presented to be targeted
TCR-peptide binding affinity	Predicted interaction strength with naïve T-cells	Stronger binding → better response
Surface presentation likelihood	Probability of actual presentation	Processing/transport affects this
Peptide concentration	Abundance on cell surface	Higher = stronger immune signal
Conservation (co-evolution)	How essential/conserved across cancer subtypes	Harder for cancer to escape via mutation
Clonality (phylogenetic position)	Present in founder clone vs subclone	Truncal mutations = all cells targeted
Specificity (cross-reactivity)	Risk of targeting normal tissue	Safety — avoid autoimmunity

Consolidated Architecture Diagram

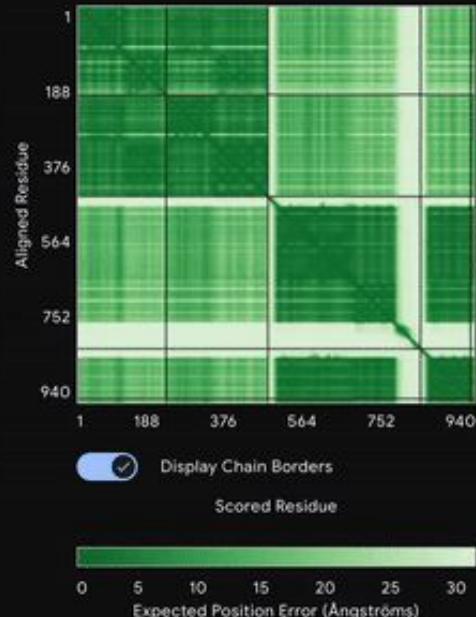
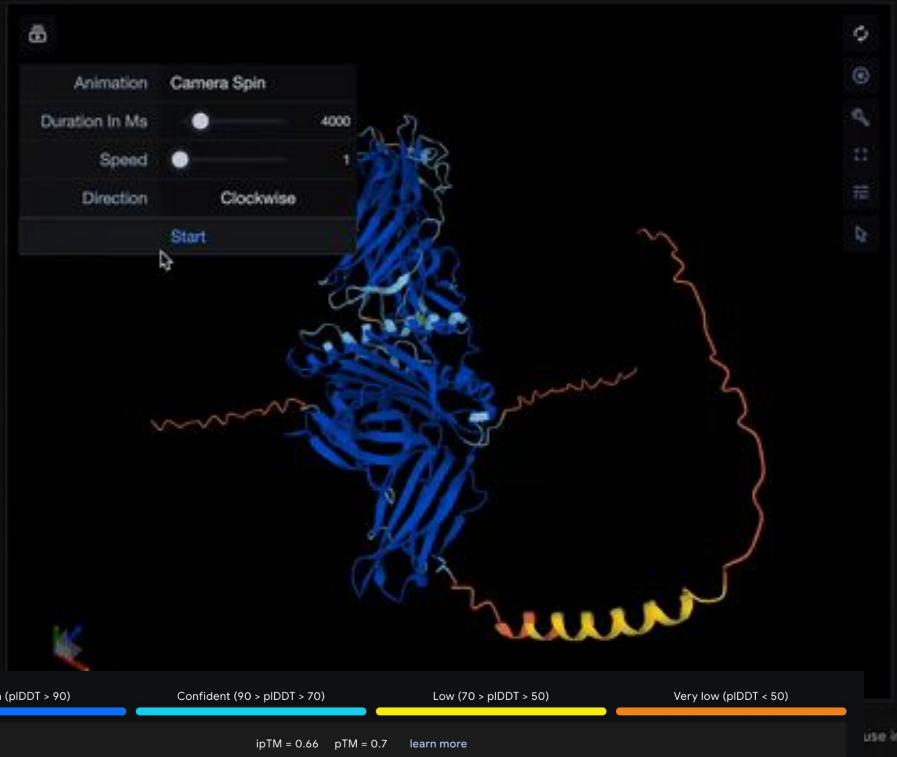


PoC: Deep Learning Structure Prediction

AlphaFold Server Server About FAQ & Guides ▾

ipTM = 0.66 pTM = 0.7 [learn more](#)

```
for i, row in df.iterrows():
    job_json = [
        {
            "name": f"TCR_MHC_job_{i+1}",
            "modelSeeds": [],
            "sequences": [
                {"proteinChain": {"sequence": row["tcr_alpha"], "count": 1}},
                {"proteinChain": {"sequence": row["tcr_beta"], "count": 1}},
                {"proteinChain": {"sequence": row["mhc_heavy_chain"], "count": 1}},
                {"proteinChain": {"sequence": row["beta_2_microglobulin"], "count": 1}},
                {"proteinChain": {"sequence": row["peptide"], "count": 1}}
            ],
            "dialect": "alphafoldserver",
            "version": 1
        }
    ]
    with open(f"Human_AF3_inputs/af3_job_{i}.json", "w") as f:
        json.dump(job_json, f, indent=2)
print("✓ Generated AlphaFold 3 JSON files (AlphaFold Server format).")
```



use in docking or scoring



Thank you