Developing Machine Learning Algorithm to Improve Kidney Transplantation Matching

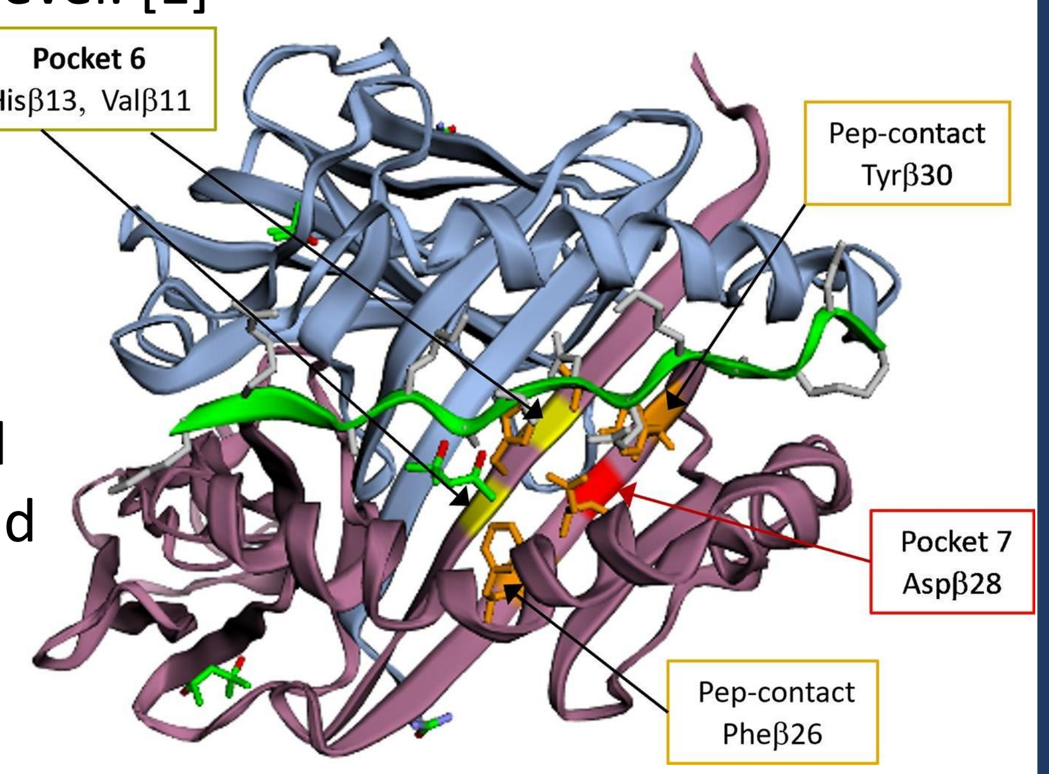
Sphia Sadek[a], Dr. Malek Kamoun[b], Dr. Ryan J. Urbanowicz[c], Harsh Bandhey[c]

College of Arts and Sciences 2025, University of Pennsylvania[a], Department of Pathology and Laboratory Medicine, University of Pennsylvania[b], Cedars Sinai Medical Center[c]

This work was supported by the University of Pennsylvania and the Penn Undergraduate Research Mentorship (PURM) program.

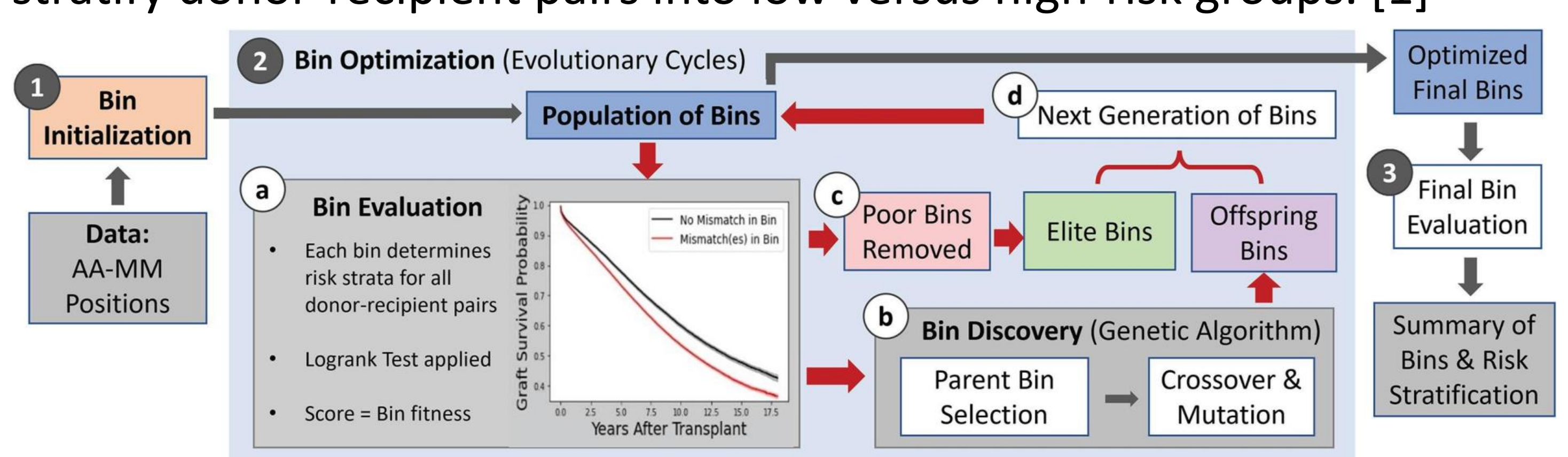
Background

- Traditional methods of kidney matching relied on antigen-level **mismatches (MMs)** in **HLA (Human Leukocyte Antigen)** molecules however fail to account for the variability at the **amino acid (AA)** level. [1]
- Polymorphic **AA** residues in the HLA molecule exhibit significant diversity at peptide binding sites as illustrated in the adjacent molecular model of **HLA**-DR molecule.
- The DR alpha and beta chain are mapped onto a ribbon diagram, colored in blue and pink, respectively.



Introduction

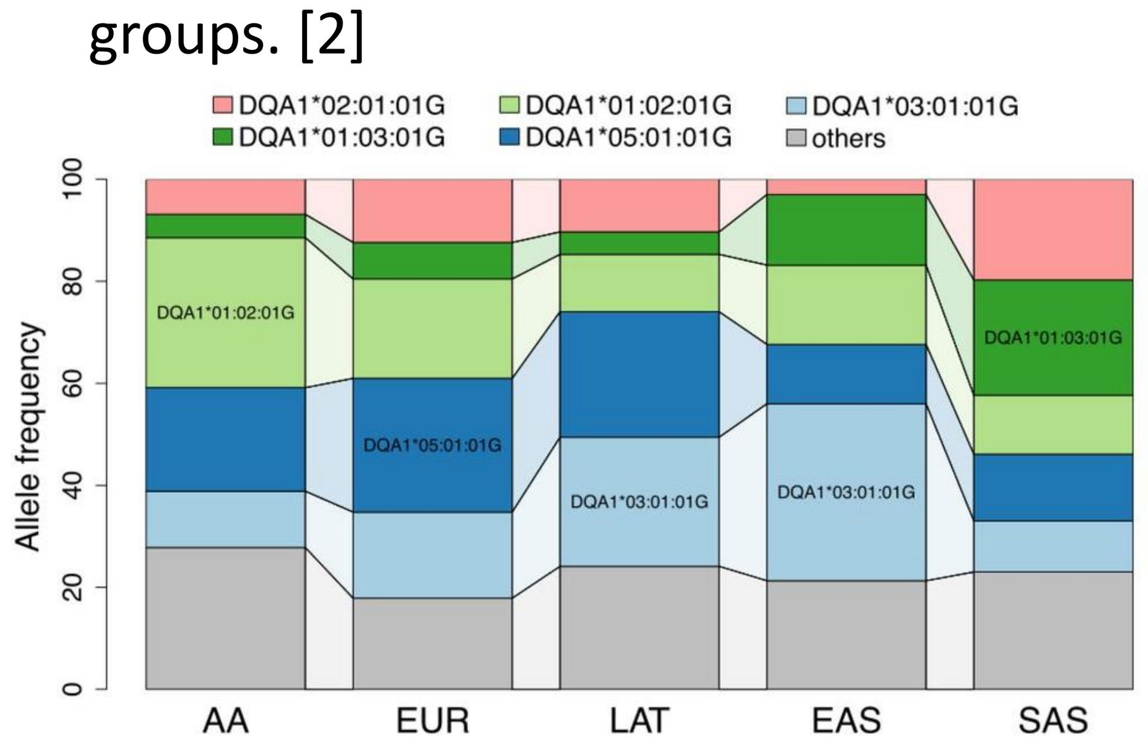
- Our research focuses on identifying risk factors of graft failure for kidney transplants.
- Previous algorithms such as HLAMatch maker, EMS3D, and HLA-EMMA, used domain knowledge. We are looking to use a **holistic data driven approach**.
- **FIBERS** (Feature Inclusion Bin Evolver for Risk Stratification) is a machine learning algorithm that discovers bins of features (**HLA AA MMs**) that will stratify donor-recipient pairs into low versus high-risk groups. [1]



Objective

FIBERS used a 0 **risk stratification threshold**: instances with 0 **feature MMs** are low risk, and 1+ **MMs** are high risk, however…

- Assuming a threshold of 0 **AA-MMs** is not a **holistic data approach**.
- **HLA** allele frequencies **vary between different ethnicities**. Thus, the frequency of the low risk 0 **AA-MMs** may be different among different ethnic groups. [2]



Allele Diversity of **HLA** DQA1.
AA: Admixed African
EUR: European
LAT: Latino
EAS: East Asian
SAS: South Asian
Common DQA1 alleles varies among populations.

Is 0 feature MMs truly the best threshold for risk stratification?

We explored the concept of having an **adaptable** risk stratification threshold that differs between bins.

Methods

1. Redefined a Bin from a Feature List to an Object.

BIN Object

Name	Feature List	Score	Threshold	Seen
"Bin_2"	["F_1", "F_2", "F_3"]	log rank score: 3421.6	2	True

2. Adapted existing simulated right-censored data, originally used to evaluate the efficacy of **FIBERS** [3], to now support user-defined risk thresholds for identifying 'low-risk' versus 'high-risk' instances.

Sample Simulated Data

Instances	P_1	P_2	P_3	…	P_10	R_1	R_2	R_3	…	R_45	True Risk	Duration
0	0	0	0	…	0	1	0	0	…	0	0	1.828136
2	0	0	0	…	0	0	0	1	…	1	0	1.295741
…	…	…	…	…	…	…	…	…	…	…	…	…
1	0	0	1	…	0	1	0	1	…	1	1	0.682862
7	0	1	1	…	1	0	1	0	…	1	1	1.187730

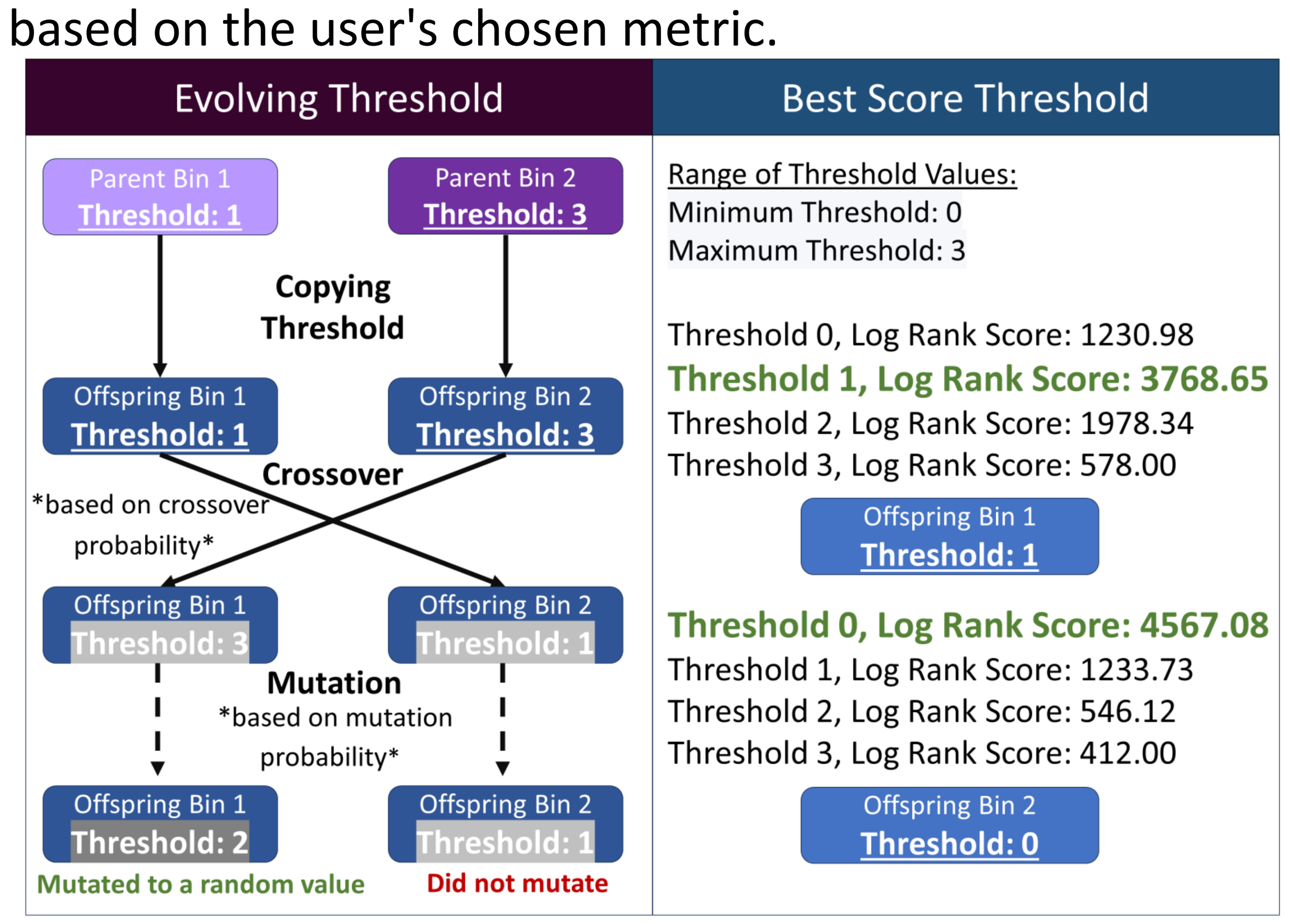
- **P_# (Predictive Features):** These convey risk information, with low risk versus high risk determined by **MMs** according to a user-defined threshold. The goal is for **FIBERS** to predict these features and stratify instances based on the **MMs**.
- **R_# (Random Features):** These do not contain risk information. **MMs** are generated based on a user-defined frequency that has no connection to risk. These features introduce randomness but do not affect risk assessment.

Low Risk Group Difference between Threshold 0 Data vs. Threshold 2 Data

Instances	P_1	P_2	P_3	P_4	P_5	
0	0	0	0	0	0	0 MMs
3	0	0	0	0	0	0 MMs
…	…	…	…	…	…	
10000	0	0	0	0	0	0 MMs

Instances	P_1	P_2	P_3	P_4	P_5	
0	0	0	0	0	0	0 MMs
3	1	0	0	0	1	2 MMs
…	…	…	…	…	…	
10000	0	1	0	0	0	1 MMs

3. Implemented two **adaptable risk stratification threshold** methods:
   1. **Evolving Threshold:** Initially assigning random threshold values within user-defined range for each bin, following evolutionary algorithms, threshold converges to the optimal value.
   2. **Best Score Threshold:** Find the best threshold for each bin by iteratively testing values within the user-defined range, using the highest fitness score based on the user's chosen metric.



4. Introduced a new hybrid approach, evolving_probability hyperparameter.

Evolving Probability: The likelihood that the iteration will *evolve the threshold.* Otherwise, the threshold value is assigned deterministically by best score metric.

Results

Simulated Data Parameters:

- 10,000 Instances
- 750 Total Features
- 10 Predictive Features
- High Risk **MMs** Frequency 40%-50%

FIBERS Parameters:

- 1,000 Iterations
- 40% mutation probability
- 80% crossover probability
- 50 bins
- min_threshold: 0, max_threshold: 3

Evolving Probability 0%

Threshold	0	1	2	3
Bin Discovery (Time in Minutes)	5:07	14:45	15:25	17:00
Iteration #	345	869	825	952
# Predictive Features	7/10	8/10	9/10	8/10
Bin Threshold	0	1	2	2
Total Time	16:52	17:13	18:43	17:54
Accuracy	100%	100%	100%	88.09%

Evolving Probability 50%

Threshold	0	1	2	3
Bin Discovery (Time in Minutes)	10:32	7:50	10:42	8:04
Iteration #	982	665	825	676
# Predictive Features	7/10	8/10	8/10	8/10
Bin Threshold	0	1	2	3
Total Time	10:43	12:19	13:09	12:20
Accuracy	100%	100%	93.14%	76.84%

Evolving Probability 100%

Threshold	0	1	2	3
Bin Discovery (Time in Minutes)	5:11	7:00	7:48	7:03
Iteration #	741	991	990	988
# Predictive Features	7/10	7/10	6/10	6/10
Bin Threshold	0	1	1	1
Total Time	6:50	7:03	7:51	7:08
Accuracy	100%	98.4%	88.8%	81.5%

How is Accuracy Measured?

- Instances are classified as low or high risk based on the threshold value of the top bin and the feature list within that bin, which is used to count the number of **MMs**.
- Accuracy is calculated by comparing these classifications to the dataset's true risk values.

Why does accuracy decrease with higher threshold values?

- Because the top bin only contains a subset of predictive features (presented in the # Predictive Features row), increasing threshold values leads to more instances placed in low risk because less features to hit the **MMs** threshold.
- Once **FIBERS** finds all 10 predictive features, accuracy reaches 100%.

Conclusion

- **FIBERS** achieved high accuracy in risk stratifying instances.
- 0% evolving probability consistently delivered the highest accuracy.
- However, using 0% evolvable probability consumes more time compared to higher values.
  - **For consistent results**, lower percentages.
  - **For time efficiency,** higher percentages.
  - Hyperparameter tuning techniques help balance both strengths.
- We are exploring the use of merging bins for faster feature discovery, which we anticipate will lead to higher accuracy.

References

Code

https://github.com/UrbsLab

Contact

sphia@sas.upenn.edu