Supporting material for A fluorescent sensor array based on carbon dots for the accurate determination of pH

Haobo Guo[a,b], Pooria Lesani[a,c,d,e], Hala Zreiqat[a,c], Elizabeth J. New[b,c,f*]

a. School of Biomedical Engineering, The University of Sydney, Sydney, NSW 2006, Australia.
b. School of Chemistry, The University of Sydney, Sydney, NSW 2006, Australia.
c. The University of Sydney Nano Institute, Sydney, NSW, 2006, Australia
d. School of Science, STEM College, RMIT University, Melbourne, VIC, 3000, Australia
e. Koch Institute for Integrative Cancer Research, Massachusetts Institute of Technology, Cambridge, MA, 02139, United States of America
f. Australian Research Council Centre of Excellence for Innovations in Peptide and Protein Science, The University of Sydney, Sydney, NSW, 2006, Australia.

elizabeth.new@sydney.edu.au

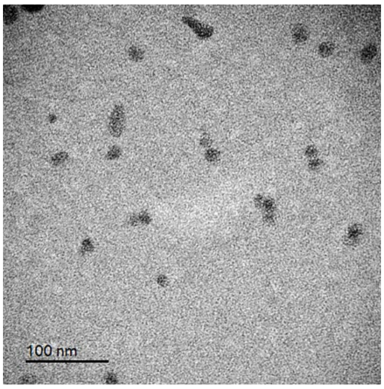
Table of Contents

List of supporting figures

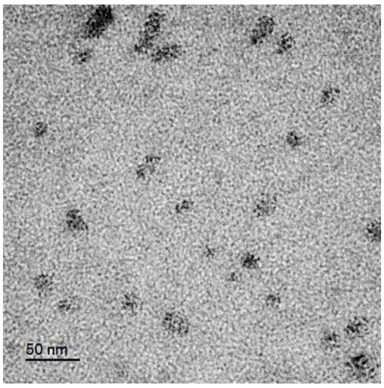
List of supporting tables

1 Characterisation of carbon dots

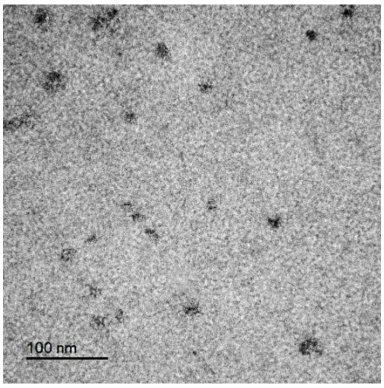
1.1 TEM of synthesised carbon dots



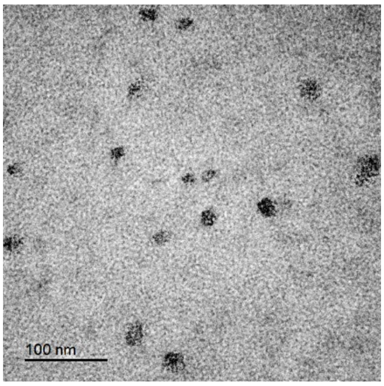
CD100



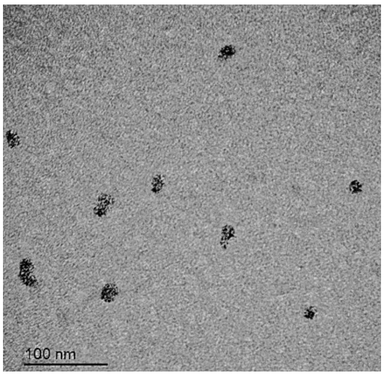
CD92



CD69



CD43



CD25

Figure S1: TEM of synthesised carbon dots

Table S1: Size distribution of CDs based on TEM analysis.

CD	Average diameter / nm	Standard deviation / nm	Comments
CD100	10.72	6.38	
CD92	7.95	3.14	
CD69	10.71	4.19	Size distribution shows two peaks (6 and 14 nm)
CD43	11.06	4.94	Broad range of particle sizes, most between 5 and 10 nm
CD25	15.45	5.35	

1.2 ATR characterisation of carbon dots

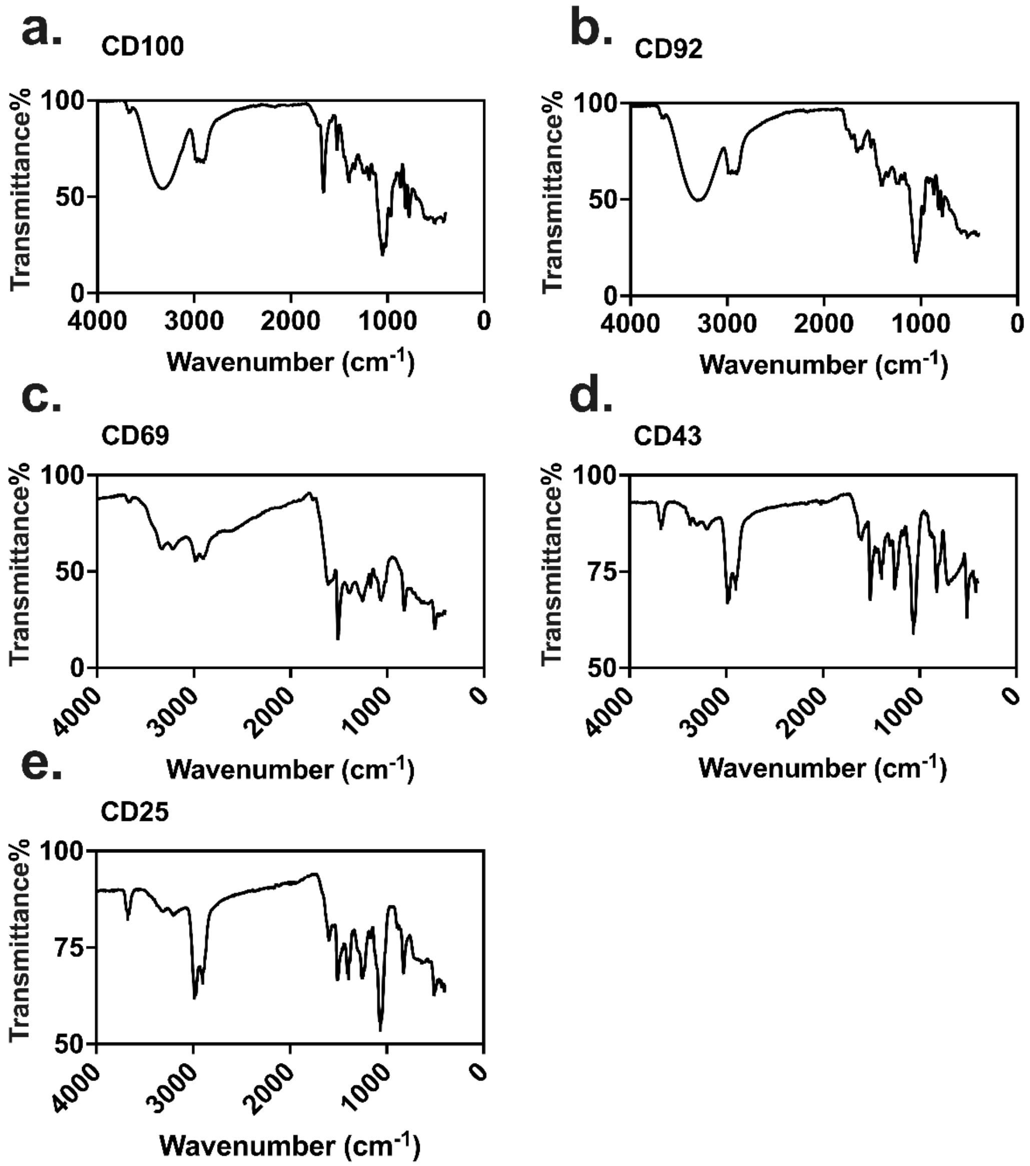


Figure S2: Attenuated total reflectance (ATR) spectra of the five carbon dots (a. CD100, b. CD92, c. CD69, d. CD43, and e. CD25).

1.3 XPS characterisation of carbon dots

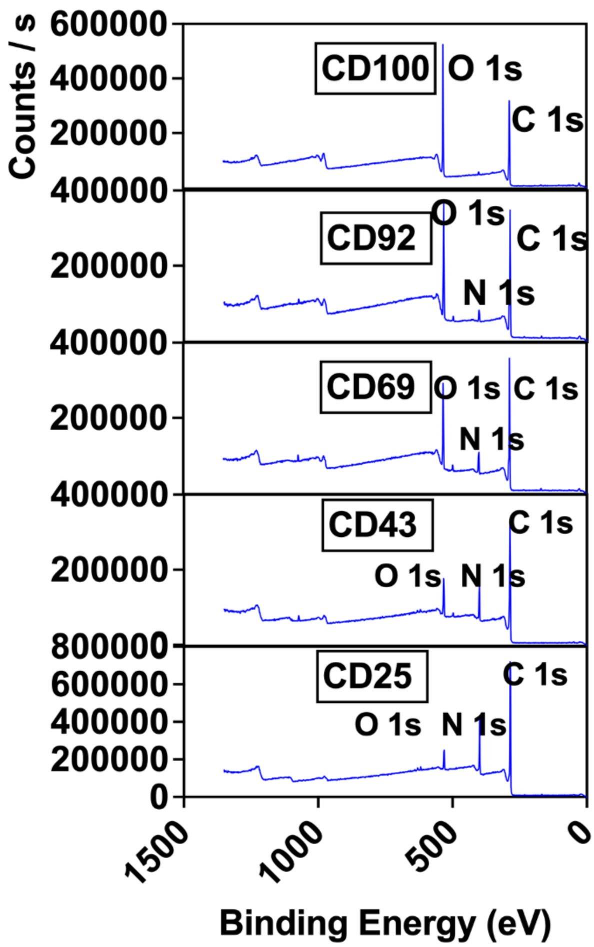


Figure S3: XPS spectra of the five CDs, in order of CD100, CD92, CD69, CD43 and CD25 from top to bottom.

Table S2: Summary of X-ray photoelectron spectroscopy (XPS data) for carbon dots.

Carbon dot	Orbital	Peak BE[a] / eV	FWHM[b] / eV	Area (P) / CPS.eV	Atomic %
CD100	C1s	285.89	3.46	1101342.77	67.51
	O1s	532.97	2.73	1281038.55	32.49
CD92	C1s	286.12	2.9	1070425.44	68.22
	N1s	401.78	3.77	133969.56	5.5
	O1s	533.28	3.24	996332.28	26.28
CD69	C1s	285.93	2.74	1078439.21	70.62
	N1s	400.77	3.93	220901.22	9.32
	O1s	532.9	3.2	740176.29	20.05
CD43	C1s	285.87	3.41	1104498.22	75.25
	N1s	400.13	3.21	311061.83	13.65
	O1s	532.74	3.59	393557.56	11.09
CD25	C1s	284.84	2.85	2140792.94	77.07
	N1s	399.03	2.78	738685.36	17.13
	O1s	531.36	3.23	389977.55	5.81

a. BE = binding energy
b. FWHM = full width at half maximum\

Table S3: Summary of XPS data of the carbon region (C 1s) for carbon dots

Carbon dot	Functional group	Peak BE / eV	FWHM / eV	Area (P) / CPS.eV	Atomic %
CD100	C-C/C=C	284.8	1.3	61871.39	43.88
	C-O	286.33	1.37	67489.98	47.91
	O-C=O	287.89	1.2	11562.66	8.22
CD92	C-C/C=C	284.8	1.53	100684.8	71.85
	C-O / C-N	286.65	1.65	18782.78	13.42
	O-C=O	288.82	1.53	20591.59	14.73
CD69	C-C/C=C	284.8	1.45	76479.39	53.59
	C-O / C-N	285.46	2.8	50598.77	35.47
	O-C=O	288.72	1.85	15583.86	10.95
CD43	C-C/C=C	284.8	1.53	23897.45	16.31
	C-O / C-N	285.94	2.65	114934.7	78.5
	O-C=O	289.36	3.37	7574.72	5.19
CD25	C-C/C=C	284.8	0.92	36832.34	14.13
	C-O / C-N	285.53	2.03	217490.85	83.48
	O-C=O	288.16	1.4	6229.01	2.39

Table S4: Summary of XPS data of the oxygen region (O 1s) for carbon dots

Carbon dot	Functional group	Peak BE / eV	FWHM / eV	Area (P) / CPS.eV	Atomic %
CD100	-OH	532.77	1.63	138287.57	85.04
	=O	533.26	2.07	24319.89	14.96
CD92	-OH	531.61	1.45	19714.04	14.63
	=O	532.51	2.51	114910.67	85.37
CD69	-OH	531.36	1.41	19107.95	18.51
	=O	532.37	2.54	84049.2	81.49
CD43	-OH	532.65	3.08	52160.32	100
CD25	-OH	531.52	1.41	11596.08	23.43
	=O	532.53	2.76	37861.02	76.57

Table S5: Summary of XPS data of the nitrogen region (N 1s) for carbon dots

Carbon dot	Functional group	Peak BE / eV	FWHM / eV	Area (P) / CPS.eV	Atomic %
CD92	Amino N	400.78	2.13	8525.03	50.73
	Graphitic N	402.64	1.82	8270.06	49.27
CD69	Amino N	400.13	1.71	16827.02	60.14
	Graphitic N	402.16	1.93	11138.72	39.86
CD43	Amino N	399.79	2.67	41330.05	100
CD25	Amino N	398.92	1.63	105106.06	100

Figure S4: Linear relationship between the atomic percentage of oxygen in hydroxyl group (vs. nitrogen in amino group) in CDs and the pH of maximum fluorescence intensity (characterised by XPS; r2 = 0.9853).

1.4 Fluorescence spectra of carbon dots under different pH conditions

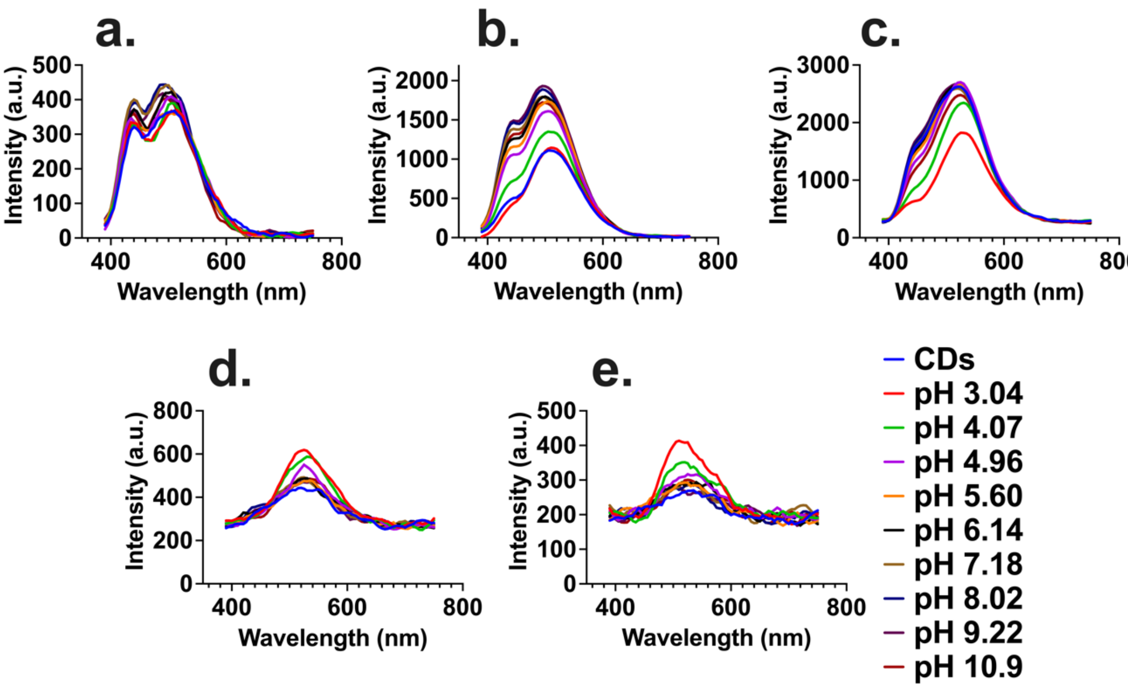


Figure S5: Fluorescence spectra of aqueous solutions of carbon dots (1 mg/mL) in different pH conditions. (a) CD100, (b) CD92, (c) CD69, (d) CD43, and (e) CD25.

Table S6: Photophysical properties of aqueous solutions of carbon dots (1 mg/mL) in different pH conditions.

Carbon dot		**CDs only**	**pH 3.04**	**pH 4.07**	**pH 4.96**	**pH 5.60**	**pH 6.14**	**pH 7.18**	**pH 8.02**	**pH 9.22**	**pH 10.9**
CD100	Emission maximum / nm	500	495	520	505	510	490	495	485	480	505
	Excitation maximum / nm	355	360	360	365	375	370	390	395	395	400
	Stokes shift / nm	145	135	160	140	135	120	105	90	85	105
CD92	Emission maximum / nm	510	515	505	505	500	500	490	500	500	500
	Excitation maximum / nm	395	370	385	370	370	385	370	375	370	370
	Stokes shift / nm	115	145	120	135	130	115	120	125	130	130
CD69	Emission maximum / nm	525	530	525	515	525	525	515	525	510	525
	Excitation maximum / nm	370	375	370	370	370	370	365	370	365	370
	Stokes shift / nm	155	155	155	145	155	155	150	155	145	155
CD43	Emission maximum / nm	535	520	530	525	535	525	515	535	530	525
	Excitation maximum / nm	365	380	370	370	375	370	370	365	350	350
	Stokes shift / nm	170	140	160	155	160	155	145	170	180	175
CD25	Emission maximum / nm	525	520	520	520	555	560	505	540	540	510
	Excitation maximum / nm	370	380	370	370	370	370	375	350	370	365
	Stokes shift / nm	155	140	150	150	185	190	130	190	170	145

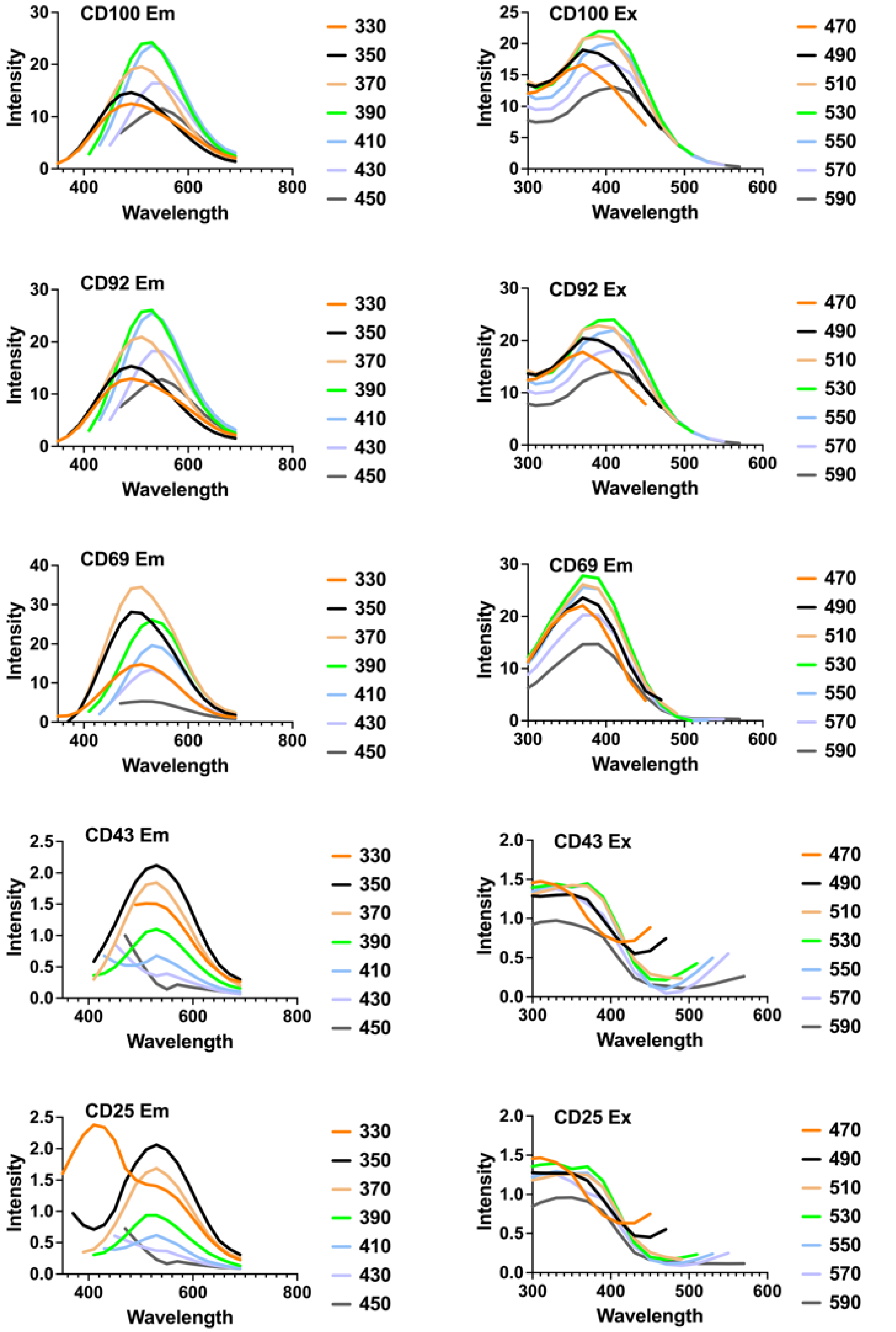


Figure S6: Fluorescence excitation and emission spectra of synthesized CDs (with λ_{ex} 330-450 nm, and λ_{em} 470-590 nm.)

1.5 Carbon dots' pH sensing stability

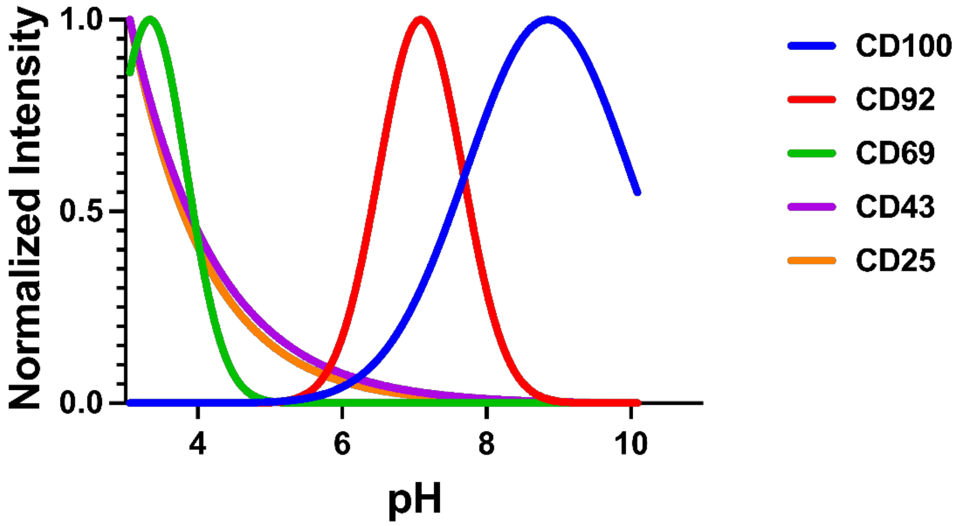


Figure S7: Over view of CDs (three months after solution prepared) with variable pH response patterns (1 mg/mL CDs; λex = 370 nm, λem integrated across 390–750 nm) in buffered Milli-Q water (25 mM buffers, pH 3–10): pH responses of individual CDs: CD100 (blue), CD92 (red), CD69 (green), CD43 (purple), and CD25 (orange), respectively.

1.6 Carbon dots pH recovery test.

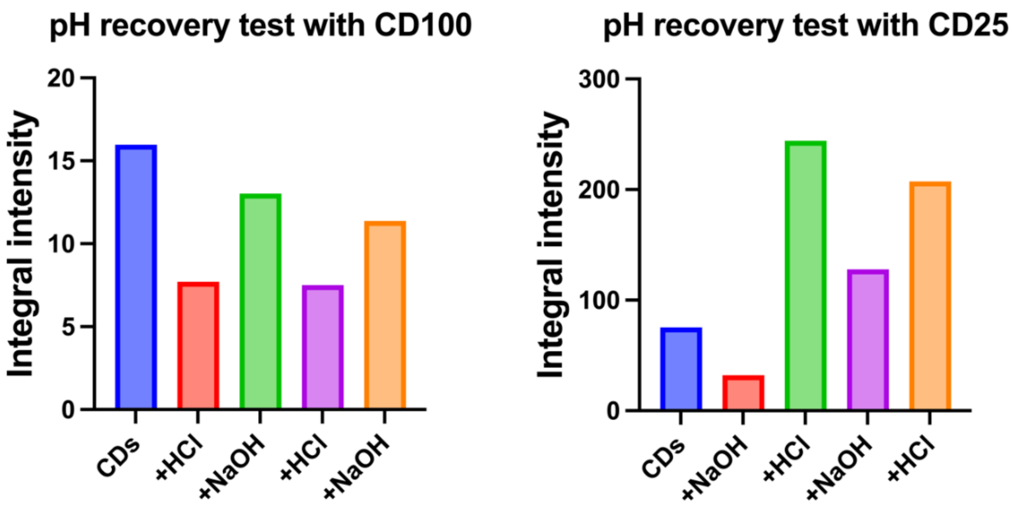


Figure S8: pH recovery test with CD100 (left) and CD25 (right) (10 mL 1 mg/mL CDs solution, pH adjusted by 5 M NaOH and HCl).

1.7 Absorption spectra of carbon dots at different pH values

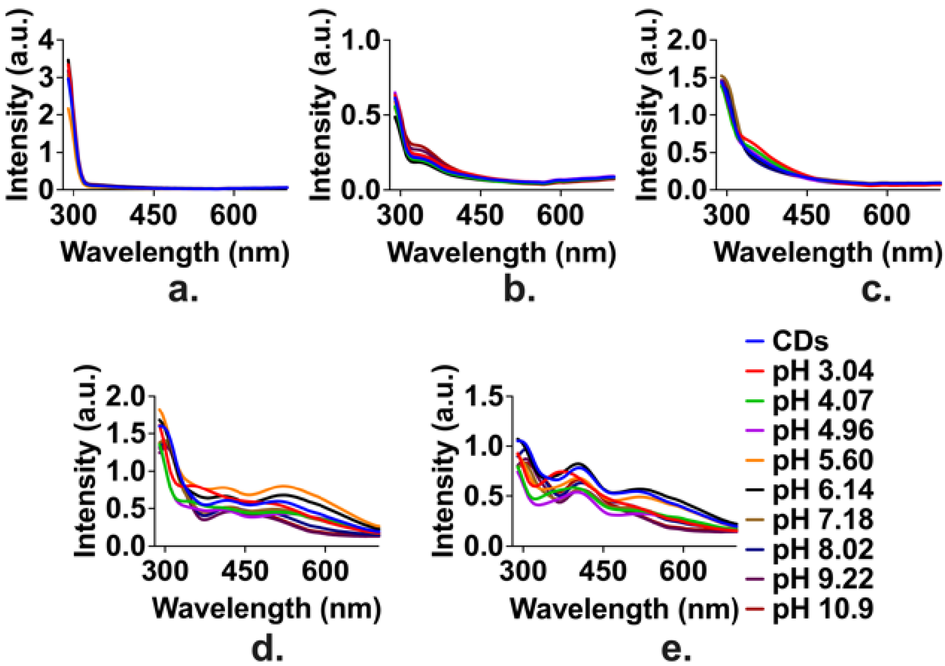


Figure S9: UV-vis absorption of five CDs under different pH buffers (a. CD100, b. CD92, c. CD69, d. CD43, and e. CD25, f. summary of CDs, at 0.25 mg/mL respectively.)

1.8 Naked eye visual observation of carbon dots pH probes under varying pH

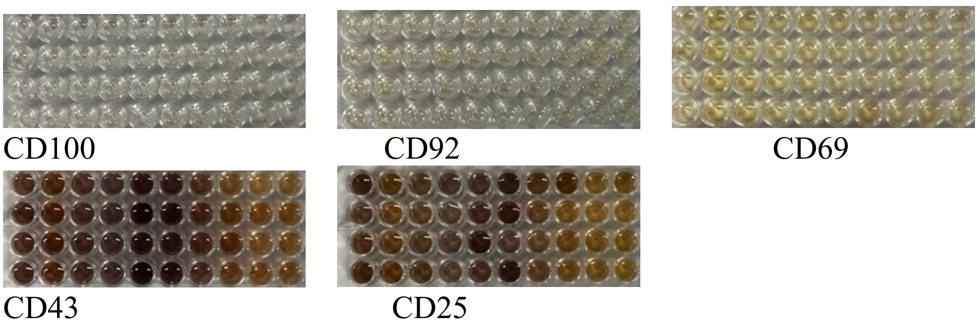


Figure S10: Naked eye visual observation of CDs pH probes under varying pH conditions (Each photo contains four replicates, with each row representing one replicate. From left to right, the columns represent CDs in MilliQ water, followed by CDs exposed to pH levels 3.04, 4.07, 4.96, 5.60, 6.14, 7.18, 8.02, 9.22, and 10.089).

2. Nine CDs based array overview

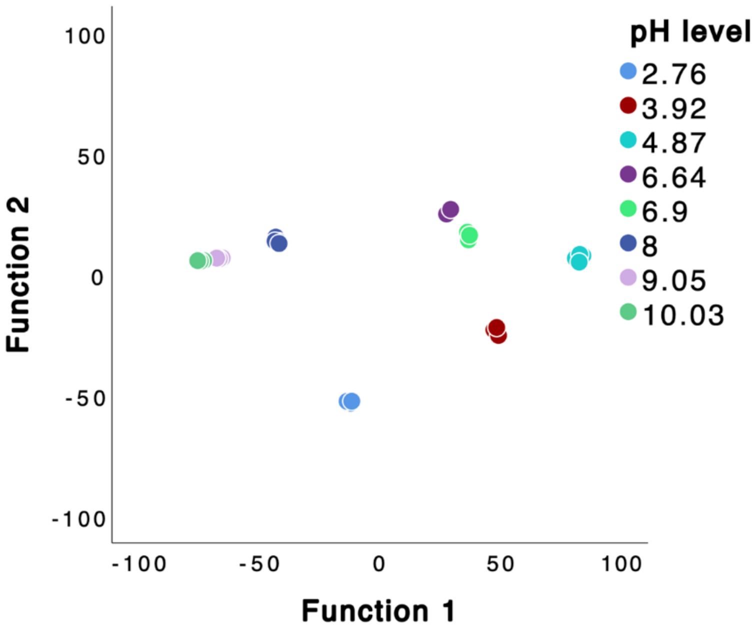


Figure S11: The LDA classification results by using nine CDs based array as pH sensor array to discriminate 8 different pH environments.

Table S7: LDA classification results by using nine CDs as pH sensor array

Classification Results[a,c]											
Items		pH	Predicted Group Membership								Total
			2.8	3.9	4.9	6.6	6.9	8	9.1	10	
Original	Count	2.76	4	0	0	0	0	0	0	0	4
		3.92	0	4	0	0	0	0	0	0	4
		4.87	0	0	4	0	0	0	0	0	4
		6.64	0	0	0	4	0	0	0	0	4
		6.9	0	0	0	0	4	0	0	0	4
		8	0	0	0	0	0	4	0	0	4
		9.05	0	0	0	0	0	0	4	0	4
		10	0	0	0	0	0	0	0	4	4
	%	2.76	100	0	0	0	0	0	0	0	100
		3.92	0	100	0	0	0	0	0	0	100
		4.87	0	0	100	0	0	0	0	0	100
		6.64	0	0	0	100	0	0	0	0	100
		6.9	0	0	0	0	100	0	0	0	100
		8	0	0	0	0	0	100	0	0	100
		9.05	0	0	0	0	0	0	100	0	100
		10	0	0	0	0	0	0	0	100	100
Cross-validated[b]	Count	2.76	4	0	0	0	0	0	0	0	4
		3.92	0	4	0	0	0	0	0	0	4
		4.87	0	0	4	0	0	0	0	0	4
		6.64	0	0	0	4	0	0	0	0	4
		6.9	0	0	0	0	4	0	0	0	4
		8	0	0	0	0	0	4	0	0	4
		9.05	0	0	0	0	0	0	4	0	4
		10	0	0	0	0	0	0	0	4	4
	%	2.76	100	0	0	0	0	0	0	0	100
		3.92	0	100	0	0	0	0	0	0	100
		4.87	0	0	100	0	0	0	0	0	100
		6.64	0	0	0	100	0	0	0	0	100
		6.9	0	0	0	0	100	0	0	0	100
		8	0	0	0	0	0	100	0	0	100
		9.05	0	0	0	0	0	0	100	0	100
		10	0	0	0	0	0	0	0	100	100

a. 100.0% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 100.0% of cross-validated grouped cases correctly classified.

3. MATLAB based LDA and SVM

3.1 MATLAB code development

Here we introduce the development of MATLAB based array classification and optimisation program. This MATLAB code was designed to develop and evaluate both support vector machine (SVM) and linear discriminant analysis (LDA) models for classifying pH levels based on feature data.

Data loading and preprocessing: The code began by clearing the workspace and setting up directories and files for storing results. It then loaded the dataset from an Excel file and displayed a message indicating the successful data load:

```
clear;
%% Load data
files = strcat('Data.xlsx');
mkdir result
fileID = fopen('result/SVM_LDA_result.txt','w');
[data,txt,raw] = xlsread(files); % load data
disp('Data loaded successfully');
disp(['Size of data: ', num2str(size(data))]);

% Extract pH levels and features
pH_levels = txt(2:end, 1); % Assuming the first row is the header
features = data(:, 1:end); % Features are from the first column to the last column
```

The feature data were then normalised, and the normalised data were saved to a file:

```
% Normalize the feature data
normalized_features = normalize(features);

% Save the normalized data to a file
normalized_data_filename = 'result/normalized_data.xlsx';
normalized_data = [pH_levels, num2cell(normalized_features)];
writecell([txt(1, :); normalized_data], normalized_data_filename);
disp(['Normalized data saved to ', normalized_data_filename]);
```

Data conversion and cross-validation setup: The pH levels were converted to categorical labels, and a 4-fold cross-validation setup was prepared:

```
% Convert pH levels to categorical labels
[pH_levels, ~, labels] = unique(pH_levels);

% Determine the number of unique pH levels
numClasses = length(unique(labels));

indices = crossvalind('Kfold', labels, 4); % Develop 4-fold cross-validation
```

Model training and performance evaluation: For each combination of features, the script trained SVM and LDA models and evaluated their performance using cross-validation. The performance metrics were stored in a table and written to a result file:

```
% Initialize a table to store performance metrics
columnNames = {'FeatureSet', 'SVM_Accuracy', 'SVM_Precision', 'SVM_Recall', 'SVM_F1', 'LDA_Accuracy',
'LDA_Precision', 'LDA_Recall', 'LDA_F1'};
perfMetrics = table('Size', [0, 9], 'VariableTypes', {'string', 'double', 'double', 'double', 'double', 'double', 'double',
'double', 'double'}, 'VariableNames', columnNames);

for feature_num = 1:9 % Assuming 9 features to represent the CDs
   disp(['Processing feature number: ', num2str(feature_num)]);
   C = nchoosek(1:9, feature_num); % Combinations of features
   for i = 1:size(C, 1)
      disp(['Processing combination: ', num2str(i), ' of feature_num: ', num2str(feature_num)]);
      feature_select = C(i,:);
      feature_data = normalized_features(:, feature_select); % Select features

      % Initialize variables to accumulate metrics
      metrics_svm = zeros(4, 4); % 4 metrics (accuracy, precision, recall, f1) for each fold
      metrics_lda = zeros(4, 4);
      for k = 1:4 % 4-fold cross-validation
         disp(['Cross-validation fold: ', num2str(k)]);
         test = (indices == k);
         train = ~test;
         train_data = feature_data(train, :);
         train_label = labels(train);
         test_data = feature_data(test, :);
         test_label = labels(test);

         % Train SVM model using fitcecoc for multi-class classification
         svm_model = fitcecoc(train_data, train_label, 'Learners', templateSVM('KernelFunction', 'linear',
'BoxConstraint', 1));
         predicted_label_svm = predict(svm_model, test_data);
         [acc_svm, prec_svm, rec_svm, f1_svm] = calculateMetrics(predicted_label_svm, test_label,
numClasses);
         metrics_svm(k, :) = [acc_svm, prec_svm, rec_svm, f1_svm];
         disp(['SVM accuracy on fold ', num2str(k), ': ', num2str(acc_svm)]);

         % Train LDA model with 'pseudoLinear' type
         lda_model = fitcdiscr(train_data, train_label, 'DiscrimType', 'pseudoLinear');
         predicted_label_lda = predict(lda_model, test_data);
         [acc_lda, prec_lda, rec_lda, f1_lda] = calculateMetrics(predicted_label_lda, test_label, numClasses);
         metrics_lda(k, :) = [acc_lda, prec_lda, rec_lda, f1_lda];
         disp(['LDA accuracy on fold ', num2str(k), ': ', num2str(acc_lda)]);
      end
      disp(['Completed processing feature set: ', num2str(feature_select)]);
      % Calculate average metrics over folds
      avgMetrics_svm = mean(metrics_svm, 1);
      avgMetrics_lda = mean(metrics_lda, 1);
      % Store in the performance metrics table
      newRow = {mat2str(feature_select), avgMetrics_svm(1), avgMetrics_svm(2), avgMetrics_svm(3),
avgMetrics_svm(4), avgMetrics_lda(1), avgMetrics_lda(2), avgMetrics_lda(3), avgMetrics_lda(4)};
      perfMetrics = [perfMetrics; newRow];

      % Write performance metrics to file after each feature set processing
      fprintf(fileID, 'Feature set: %s, SVM: Acc: %f, Prec: %f, Rec: %f, F1: %f, LDA: Acc: %f, Prec: %f,
Rec: %f, F1: %f\n', ...
            perfMetrics.FeatureSet{end}, perfMetrics.SVM_Accuracy(end), perfMetrics.SVM_Precision(end),
perfMetrics.SVM_Recall(end), perfMetrics.SVM_F1(end), ...
            perfMetrics.LDA_Accuracy(end), perfMetrics.LDA_Precision(end), perfMetrics.LDA_Recall(end),
perfMetrics.LDA_F1(end));
   end
end
```

```
fclose(fileID);
disp('Program completed.');
```

Best performing feature set identification: The script identified the best performing feature set for both SVM and LDA based on the accuracy metric:

```
% Identify the best performing combination for SVM
[~, best_svm_idx] = max(perfMetrics.SVM_Accuracy);
best_svm_features = perfMetrics.FeatureSet{best_svm_idx};
best_svm_accuracy = perfMetrics.SVM_Accuracy(best_svm_idx);
best_svm_precision = perfMetrics.SVM_Precision(best_svm_idx);
best_svm_recall = perfMetrics.SVM_Recall(best_svm_idx);
best_svm_f1 = perfMetrics.SVM_F1(best_svm_idx);

disp('Best SVM Feature Set:');
disp(['Feature Set: ', best_svm_features]);
disp(['Accuracy: ', num2str(best_svm_accuracy)]);
disp(['Precision: ', num2str(best_svm_precision)]);
disp(['Recall: ', num2str(best_svm_recall)]);
disp(['F1 Score: ', num2str(best_svm_f1)]);

% Identify the best performing combination for LDA
[~, best_lda_idx] = max(perfMetrics.LDA_Accuracy);
best_lda_features = perfMetrics.FeatureSet{best_lda_idx};
best_lda_accuracy = perfMetrics.LDA_Accuracy(best_lda_idx);
best_lda_precision = perfMetrics.LDA_Precision(best_lda_idx);
best_lda_recall = perfMetrics.LDA_Recall(best_lda_idx);
best_lda_f1 = perfMetrics.LDA_F1(best_lda_idx);

disp('Best LDA Feature Set:');
disp(['Feature Set: ', best_lda_features]);
disp(['Accuracy: ', num2str(best_lda_accuracy)]);
disp(['Precision: ', num2str(best_lda_precision)]);
disp(['Recall: ', num2str(best_lda_recall)]);
disp(['F1 Score: ', num2str(best_lda_f1)]);
```

Utility function for metric calculation: A utility function was included to calculate the accuracy, precision, recall, and F1 score for the predicted labels:

```
function [accuracy, precision, recall, f1score] = calculateMetrics(predicted, actual, numClasses)
  % Calculate confusion matrix
  cm = confusionmat(actual, predicted);

  % Initialize metrics
  classPrecision = zeros(numClasses, 1);
  classRecall = zeros(numClasses, 1);

  % Calculate precision and recall for each class
  for i = 1:numClasses
    truePositives = cm(i, i);
    falsePositives = sum(cm(:, i)) - truePositives;
    falseNegatives = sum(cm(i, :)) - truePositives;

    classPrecision(i) = truePositives / (truePositives + falsePositives);
    classRecall(i) = truePositives / (truePositives + falseNegatives);
  end
```

```
    % Handle NaN cases (if any division by zero occurs)
    classPrecision(isnan(classPrecision)) = 0;
    classRecall(isnan(classRecall)) = 0;

    % Calculate macro-averaged precision, recall, and F1 score
    precision = mean(classPrecision);
    recall = mean(classRecall);
    f1score = 2 * (precision * recall) / (precision + recall);

    % Calculate overall accuracy
    accuracy = sum(diag(cm)) / sum(cm(:));
end
```

This structured approach ensured a thorough evaluation of different feature combinations, providing insights into the optimal configuration for accurate pH level classification using SVM and LDA models.

3.2 MATLAB based array optimisation results

Table S8: The features numbers matched for carbon dots

Feature number	Carbon dot
1	CD100
2	CD100 (-OH protected)
3	CD69
4	CD69 (-OH protected)
5	CD69 (-NH_2 protected)
6	CD25
7	CD25 (-NH_2 protected)
8	CD92
9	CD43

Table S9: Summary of the optimisation of SVM and LDA using all combinations of features shown in Table S7. Acc = accuracy; Prec = precision; Rec = recall; F1 = F1 score (F-measure).

	SVM				LDA			
Feature set	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
1	0.750000	0.672619	0.750000	0.708981	0.785714	0.732143	0.785714	0.757475
2	0.250000	0.098214	0.250000	0.139751	0.214286	0.113095	0.214286	0.144558
3	0.785714	0.714286	0.785714	0.747239	0.750000	0.696429	0.750000	0.720842
4	0.428571	0.297619	0.428571	0.346918	0.500000	0.386905	0.500000	0.434917
5	0.571429	0.470238	0.571429	0.515699	0.714286	0.595238	0.714286	0.649093
6	0.571429	0.488095	0.571429	0.526084	0.642857	0.535714	0.642857	0.580525
7	0.464286	0.375000	0.464286	0.412245	0.571429	0.470238	0.571429	0.515306
8	0.714286	0.642857	0.714286	0.675604	0.750000	0.732143	0.750000	0.740683
9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 2	0.678571	0.559524	0.678571	0.612186	0.785714	0.726190	0.785714	0.754012
1 3	0.857143	0.803571	0.857143	0.829111	0.857143	0.821429	0.857143	0.838509
1 4	0.964286	0.946429	0.964286	0.954969	0.964286	0.946429	0.964286	0.954969
1 5	0.892857	0.851190	0.892857	0.870785	0.892857	0.851190	0.892857	0.870785
1 6	0.928571	0.892857	0.928571	0.909938	0.928571	0.892857	0.928571	0.909938
1 7	0.750000	0.696429	0.750000	0.721679	0.750000	0.678571	0.750000	0.712444
1 8	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
2 3	0.571429	0.410714	0.571429	0.472666	0.714286	0.678571	0.714286	0.695445
2 4	0.607143	0.455357	0.607143	0.518857	0.535714	0.458333	0.535714	0.492515

2 5	0.535714	0.389881	0.535714	0.451220	0.678571	0.565476	0.678571	0.615058
2 6	0.607143	0.508929	0.607143	0.553476	0.607143	0.547619	0.607143	0.573436
2 7	0.500000	0.392857	0.500000	0.438424	0.500000	0.434524	0.500000	0.464054
2 8	0.642857	0.511905	0.642857	0.569444	0.892857	0.839286	0.892857	0.863699
2 9	0.535714	0.410714	0.535714	0.463946	1.000000	1.000000	1.000000	1.000000
3 4	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
3 5	0.928571	0.892857	0.928571	0.909938	0.928571	0.892857	0.928571	0.909938
3 6	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
3 7	0.785714	0.750000	0.785714	0.767081	0.750000	0.732143	0.750000	0.740683
3 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
3 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
4 5	0.642857	0.523810	0.642857	0.577201	0.750000	0.660714	0.750000	0.702045
4 6	0.642857	0.494048	0.642857	0.557221	0.642857	0.494048	0.642857	0.557221
4 7	0.714286	0.625000	0.714286	0.666205	0.785714	0.678571	0.785714	0.727398
4 8	0.892857	0.857143	0.892857	0.874142	0.928571	0.892857	0.928571	0.909938
4 9	0.821429	0.732143	0.821429	0.773637	1.000000	1.000000	1.000000	1.000000
5 6	0.678571	0.580357	0.678571	0.625111	0.642857	0.541667	0.642857	0.586789
5 7	0.678571	0.583333	0.678571	0.626946	0.821429	0.750000	0.821429	0.784080
5 8	0.857143	0.797619	0.857143	0.825754	0.892857	0.851190	0.892857	0.870785
5 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
6 7	0.714286	0.625000	0.714286	0.666205	0.714286	0.642857	0.714286	0.675306
6 8	0.928571	0.892857	0.928571	0.909938	0.964286	0.946429	0.964286	0.954969
6 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
7 8	0.821429	0.732143	0.821429	0.768620	0.964286	0.946429	0.964286	0.954969
7 9	0.892857	0.857143	0.892857	0.874142	1.000000	1.000000	1.000000	1.000000
8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 3	0.714286	0.625000	0.714286	0.666622	0.857143	0.821429	0.857143	0.838509
1 2 4	0.892857	0.875000	0.892857	0.883540	0.928571	0.892857	0.928571	0.909938
1 2 5	0.714286	0.595238	0.714286	0.649093	0.821429	0.750000	0.821429	0.784080
1 2 6	0.857143	0.821429	0.857143	0.838509	0.892857	0.839286	0.892857	0.864907
1 2 7	0.678571	0.613095	0.678571	0.643663	0.714286	0.636905	0.714286	0.673141
1 2 8	0.821429	0.750000	0.821429	0.784080	0.964286	0.946429	0.964286	0.954969

1 2 9	0.821429	0.803571	0.821429	0.812112	1.000000	1.000000	1.000000	1.000000
1 3 4	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 5	0.857143	0.797619	0.857143	0.825754	0.928571	0.892857	0.928571	0.909938
1 3 6	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 7	0.821429	0.750000	0.821429	0.784080	0.821429	0.767857	0.821429	0.793478
1 3 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 3 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 4 5	0.964286	0.946429	0.964286	0.954969	0.928571	0.910714	0.928571	0.919173
1 4 6	0.964286	0.946429	0.964286	0.954969	0.928571	0.892857	0.928571	0.909938
1 4 7	0.928571	0.892857	0.928571	0.909938	0.964286	0.946429	0.964286	0.954969
1 4 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 4 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 5 6	0.928571	0.892857	0.928571	0.909938	0.928571	0.892857	0.928571	0.909938
1 5 7	0.892857	0.851190	0.892857	0.870785	0.892857	0.851190	0.892857	0.870785
1 5 8	0.892857	0.839286	0.892857	0.864907	0.964286	0.946429	0.964286	0.954969
1 5 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 6 7	0.928571	0.892857	0.928571	0.909938	0.928571	0.892857	0.928571	0.909938
1 6 8	0.964286	0.946429	0.964286	0.954969	0.964286	0.946429	0.964286	0.954969
1 6 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 7 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 7 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
2 3 4	0.714286	0.607143	0.714286	0.655763	1.000000	1.000000	1.000000	1.000000
2 3 5	0.642857	0.517857	0.642857	0.572549	0.857143	0.821429	0.857143	0.838509
2 3 6	0.785714	0.714286	0.785714	0.747032	1.000000	1.000000	1.000000	1.000000
2 3 7	0.714286	0.625000	0.714286	0.666622	0.714286	0.714286	0.714286	0.714286
2 3 8	0.750000	0.642857	0.750000	0.691117	1.000000	1.000000	1.000000	1.000000
2 3 9	0.821429	0.767857	0.821429	0.792063	1.000000	1.000000	1.000000	1.000000
2 4 5	0.607143	0.508929	0.607143	0.553476	0.642857	0.529762	0.642857	0.580664
2 4 6	0.607143	0.502976	0.607143	0.550012	0.607143	0.511905	0.607143	0.552022
2 4 7	0.750000	0.714286	0.750000	0.731159	0.714286	0.625000	0.714286	0.666205
2 4 8	0.750000	0.666667	0.750000	0.704417	0.928571	0.892857	0.928571	0.909938

2 4 9	0.714286	0.625000	0.714286	0.664512	1.000000	1.000000	1.000000	1.000000
2 5 6	0.607143	0.508929	0.607143	0.553476	0.571429	0.467262	0.571429	0.513212
2 5 7	0.535714	0.446429	0.535714	0.487013	0.714286	0.660714	0.714286	0.685128
2 5 8	0.607143	0.476190	0.607143	0.533498	0.857143	0.797619	0.857143	0.824546
2 5 9	0.714286	0.607143	0.714286	0.655763	1.000000	1.000000	1.000000	1.000000
2 6 7	0.607143	0.547619	0.607143	0.575331	0.607143	0.547619	0.607143	0.574546
2 6 8	0.714286	0.619048	0.714286	0.662849	0.892857	0.851190	0.892857	0.870785
2 6 9	0.750000	0.660714	0.750000	0.702001	1.000000	1.000000	1.000000	1.000000
2 7 8	0.642857	0.553571	0.642857	0.593055	0.928571	0.904762	0.928571	0.915816
2 7 9	0.821429	0.785714	0.821429	0.802713	1.000000	1.000000	1.000000	1.000000
2 8 9	0.892857	0.839286	0.892857	0.863699	1.000000	1.000000	1.000000	1.000000
3 4 5	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
3 4 6	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
3 4 7	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
3 4 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3 4 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
3 5 6	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3 5 7	0.892857	0.839286	0.892857	0.864907	0.892857	0.875000	0.892857	0.883540
3 5 8	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
3 5 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3 6 7	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
3 6 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3 6 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3 7 8	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
3 7 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
3 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
4 5 6	0.678571	0.583333	0.678571	0.626903	0.750000	0.654762	0.750000	0.698688
4 5 7	0.750000	0.696429	0.750000	0.721843	0.821429	0.732143	0.821429	0.773637
4 5 8	0.892857	0.857143	0.892857	0.874142	0.892857	0.839286	0.892857	0.864907
4 5 9	0.785714	0.714286	0.785714	0.748240	1.000000	1.000000	1.000000	1.000000
4 6 7	0.821429	0.732143	0.821429	0.773637	0.750000	0.660714	0.750000	0.701001
4 6 8	0.892857	0.857143	0.892857	0.874142	0.964286	0.946429	0.964286	0.954969

4 6 9	0.821429	0.732143	0.821429	0.773637	1.000000	1.000000	1.000000	1.000000
4 7 8	0.928571	0.892857	0.928571	0.909938	0.964286	0.946429	0.964286	0.954969
4 7 9	0.785714	0.696429	0.785714	0.737841	1.000000	1.000000	1.000000	1.000000
4 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
5 6 7	0.785714	0.714286	0.785714	0.747239	0.750000	0.672619	0.750000	0.708087
5 6 8	0.892857	0.839286	0.892857	0.864907	0.928571	0.892857	0.928571	0.909938
5 6 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
5 7 8	0.857143	0.785714	0.857143	0.818668	0.964286	0.946429	0.964286	0.954969
5 7 9	0.892857	0.839286	0.892857	0.864907	1.000000	1.000000	1.000000	1.000000
5 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
6 7 8	0.857143	0.785714	0.857143	0.817460	0.964286	0.946429	0.964286	0.954969
6 7 9	0.928571	0.892857	0.928571	0.908730	1.000000	1.000000	1.000000	1.000000
6 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 3 4	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
1 2 3 5	0.714286	0.625000	0.714286	0.666622	0.857143	0.785714	0.857143	0.819876
1 2 3 6	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 3 7	0.714286	0.625000	0.714286	0.666622	0.821429	0.767857	0.821429	0.793478
1 2 3 8	0.857143	0.785714	0.857143	0.819876	1.000000	1.000000	1.000000	1.000000
1 2 3 9	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
1 2 4 5	0.857143	0.833333	0.857143	0.844388	0.928571	0.892857	0.928571	0.909938
1 2 4 6	0.892857	0.875000	0.892857	0.883540	0.892857	0.839286	0.892857	0.864907
1 2 4 7	0.857143	0.821429	0.857143	0.838509	0.928571	0.892857	0.928571	0.909938
1 2 4 8	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 2 4 9	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
1 2 5 6	0.857143	0.833333	0.857143	0.844388	0.892857	0.839286	0.892857	0.864907
1 2 5 7	0.714286	0.625000	0.714286	0.666622	0.857143	0.803571	0.857143	0.829111
1 2 5 8	0.821429	0.750000	0.821429	0.784080	0.928571	0.892857	0.928571	0.909938
1 2 5 9	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
1 2 6 7	0.857143	0.821429	0.857143	0.838509	0.892857	0.839286	0.892857	0.864907
1 2 6 8	0.928571	0.892857	0.928571	0.909938	0.964286	0.946429	0.964286	0.954969
1 2 6 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000

1 2 7 8	0.785714	0.714286	0.785714	0.748284	0.964286	0.946429	0.964286	0.954969
1 2 7 9	0.857143	0.821429	0.857143	0.838509	1.000000	1.000000	1.000000	1.000000
1 2 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 4 5	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 4 6	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 4 7	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 3 4 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 3 4 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 5 6	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 5 7	0.857143	0.797619	0.857143	0.825754	0.928571	0.892857	0.928571	0.909938
1 3 5 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 5 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 6 7	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 3 6 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 3 6 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 7 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 7 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 3 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 4 5 6	0.964286	0.946429	0.964286	0.954969	0.892857	0.857143	0.892857	0.874142
1 4 5 7	0.928571	0.892857	0.928571	0.909938	0.964286	0.946429	0.964286	0.954969
1 4 5 8	0.964286	0.946429	0.964286	0.954969	0.964286	0.946429	0.964286	0.954969
1 4 5 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 4 6 7	0.928571	0.892857	0.928571	0.909938	0.928571	0.892857	0.928571	0.909938
1 4 6 8	1.000000	1.000000	1.000000	1.000000	0.964286	0.946429	0.964286	0.954969
1 4 6 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 4 7 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 4 7 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 4 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 5 6 7	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 5 6 8	0.964286	0.946429	0.964286	0.954969	0.928571	0.892857	0.928571	0.909938
1 5 6 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 5 7 8	0.928571	0.892857	0.928571	0.909938	0.964286	0.946429	0.964286	0.954969

1 5 7 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 5 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 6 7 8	0.964286	0.946429	0.964286	0.954969	0.964286	0.946429	0.964286	0.954969
1 6 7 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 6 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
2 3 4 5	0.785714	0.714286	0.785714	0.747032	1.000000	1.000000	1.000000	1.000000
2 3 4 6	0.785714	0.714286	0.785714	0.747032	1.000000	1.000000	1.000000	1.000000
2 3 4 7	0.857143	0.821429	0.857143	0.838509	1.000000	1.000000	1.000000	1.000000
2 3 4 8	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
2 3 4 9	0.821429	0.767857	0.821429	0.793271	1.000000	1.000000	1.000000	1.000000
2 3 5 6	0.785714	0.714286	0.785714	0.747032	1.000000	1.000000	1.000000	1.000000
2 3 5 7	0.714286	0.625000	0.714286	0.666622	0.892857	0.875000	0.892857	0.883540
2 3 5 8	0.750000	0.642857	0.750000	0.691117	1.000000	1.000000	1.000000	1.000000
2 3 5 9	0.785714	0.714286	0.785714	0.747032	1.000000	1.000000	1.000000	1.000000
2 3 6 7	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
2 3 6 8	0.892857	0.839286	0.892857	0.863699	1.000000	1.000000	1.000000	1.000000
2 3 6 9	0.857143	0.785714	0.857143	0.817460	1.000000	1.000000	1.000000	1.000000
2 3 7 8	0.785714	0.714286	0.785714	0.748284	1.000000	1.000000	1.000000	1.000000
2 3 7 9	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
2 3 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
2 4 5 6	0.642857	0.556548	0.642857	0.596251	0.678571	0.583333	0.678571	0.626946
2 4 5 7	0.750000	0.714286	0.750000	0.731203	0.714286	0.642857	0.714286	0.675729
2 4 5 8	0.750000	0.666667	0.750000	0.704417	0.821429	0.744048	0.821429	0.780723
2 4 5 9	0.714286	0.642857	0.714286	0.675397	1.000000	1.000000	1.000000	1.000000
2 4 6 7	0.785714	0.750000	0.785714	0.766999	0.678571	0.571429	0.678571	0.619967
2 4 6 8	0.821429	0.761905	0.821429	0.789808	0.892857	0.839286	0.892857	0.864907
2 4 6 9	0.750000	0.660714	0.750000	0.700794	1.000000	1.000000	1.000000	1.000000
2 4 7 8	0.857143	0.797619	0.857143	0.824546	0.964286	0.946429	0.964286	0.954969
2 4 7 9	0.821429	0.785714	0.821429	0.802713	1.000000	1.000000	1.000000	1.000000
2 4 8 9	0.892857	0.839286	0.892857	0.864907	1.000000	1.000000	1.000000	1.000000
2 5 6 7	0.642857	0.601190	0.642857	0.620785	0.678571	0.619048	0.678571	0.645975

2 5 6 8	0.750000	0.654762	0.750000	0.698688	0.857143	0.797619	0.857143	0.825754
2 5 6 9	0.714286	0.607143	0.714286	0.655763	1.000000	1.000000	1.000000	1.000000
2 5 7 8	0.678571	0.607143	0.678571	0.640674	0.892857	0.857143	0.892857	0.874142
2 5 7 9	0.857143	0.821429	0.857143	0.838509	1.000000	1.000000	1.000000	1.000000
2 5 8 9	0.892857	0.839286	0.892857	0.863699	1.000000	1.000000	1.000000	1.000000
2 6 7 8	0.785714	0.761905	0.785714	0.772959	0.964286	0.946429	0.964286	0.954969
2 6 7 9	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
2 6 8 9	0.892857	0.839286	0.892857	0.863699	1.000000	1.000000	1.000000	1.000000
2 7 8 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
3 4 5 6	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
3 4 5 7	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
3 4 5 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
3 4 5 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
3 4 6 7	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
3 4 6 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3 4 6 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
3 4 7 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3 4 7 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
3 4 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3 5 6 7	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
3 5 6 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3 5 6 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3 5 7 8	0.892857	0.839286	0.892857	0.864907	1.000000	1.000000	1.000000	1.000000
3 5 7 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
3 5 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3 6 7 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
3 6 7 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
3 6 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
4 5 6 7	0.785714	0.714286	0.785714	0.747239	0.821429	0.732143	0.821429	0.773637
4 5 6 8	0.892857	0.857143	0.892857	0.874142	0.892857	0.839286	0.892857	0.864907
4 5 6 9	0.785714	0.714286	0.785714	0.748240	1.000000	1.000000	1.000000	1.000000

4 5 7 8	0.928571	0.892857	0.928571	0.909938	0.964286	0.946429	0.964286	0.954969
4 5 7 9	0.821429	0.767857	0.821429	0.793478	1.000000	1.000000	1.000000	1.000000
4 5 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
4 6 7 8	0.928571	0.892857	0.928571	0.909938	0.964286	0.946429	0.964286	0.954969
4 6 7 9	0.857143	0.785714	0.857143	0.818668	1.000000	1.000000	1.000000	1.000000
4 6 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
4 7 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
5 6 7 8	0.928571	0.892857	0.928571	0.909938	0.964286	0.946429	0.964286	0.954969
5 6 7 9	0.892857	0.839286	0.892857	0.864907	1.000000	1.000000	1.000000	1.000000
5 6 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
5 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
6 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 3 4 5	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
1 2 3 4 6	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
1 2 3 4 7	0.857143	0.821429	0.857143	0.838509	1.000000	1.000000	1.000000	1.000000
1 2 3 4 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 2 3 4 9	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
1 2 3 5 6	0.928571	0.928571	0.928571	0.928571	1.000000	1.000000	1.000000	1.000000
1 2 3 5 7	0.714286	0.625000	0.714286	0.666622	0.892857	0.839286	0.892857	0.864907
1 2 3 5 8	0.857143	0.785714	0.857143	0.819876	1.000000	1.000000	1.000000	1.000000
1 2 3 5 9	0.928571	0.928571	0.928571	0.928571	1.000000	1.000000	1.000000	1.000000
1 2 3 6 7	0.892857	0.839286	0.892857	0.864907	1.000000	1.000000	1.000000	1.000000
1 2 3 6 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 3 6 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 2 3 7 8	0.785714	0.714286	0.785714	0.748284	1.000000	1.000000	1.000000	1.000000
1 2 3 7 9	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
1 2 3 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 4 5 6	0.892857	0.875000	0.892857	0.883540	0.892857	0.839286	0.892857	0.864907
1 2 4 5 7	0.857143	0.821429	0.857143	0.838509	0.928571	0.892857	0.928571	0.909938
1 2 4 5 8	0.928571	0.892857	0.928571	0.909938	0.964286	0.946429	0.964286	0.954969
1 2 4 5 9	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
1 2 4 6 7	0.857143	0.821429	0.857143	0.838509	0.892857	0.839286	0.892857	0.864907

1 2 4 6 8	0.928571	0.892857	0.928571	0.909938	0.964286	0.946429	0.964286	0.954969
1 2 4 6 9	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
1 2 4 7 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 4 7 9	0.892857	0.839286	0.892857	0.864907	1.000000	1.000000	1.000000	1.000000
1 2 4 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 2 5 6 7	0.857143	0.821429	0.857143	0.838509	0.928571	0.892857	0.928571	0.909938
1 2 5 6 8	0.928571	0.892857	0.928571	0.909938	0.964286	0.946429	0.964286	0.954969
1 2 5 6 9	0.928571	0.928571	0.928571	0.928571	1.000000	1.000000	1.000000	1.000000
1 2 5 7 8	0.785714	0.714286	0.785714	0.748284	0.964286	0.946429	0.964286	0.954969
1 2 5 7 9	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
1 2 5 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 6 7 8	0.964286	0.946429	0.964286	0.954969	0.964286	0.946429	0.964286	0.954969
1 2 6 7 9	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
1 2 6 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 4 5 6	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 4 5 7	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 3 4 5 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 3 4 5 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 4 6 7	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 3 4 6 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 3 4 6 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 4 7 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 3 4 7 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 3 4 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 3 5 6 7	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 3 5 6 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 5 6 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 3 5 7 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 5 7 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 5 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 6 7 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000

1 3 6 7 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 3 6 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 3 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 4 5 6 7	0.928571	0.892857	0.928571	0.909938	0.964286	0.946429	0.964286	0.954969
1 4 5 6 8	1.000000	1.000000	1.000000	1.000000	0.928571	0.892857	0.928571	0.909938
1 4 5 6 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 4 5 7 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 4 5 7 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 4 5 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 4 6 7 8	0.964286	0.946429	0.964286	0.954969	0.964286	0.946429	0.964286	0.954969
1 4 6 7 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 4 6 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 4 7 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 5 6 7 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 5 6 7 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 5 6 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 5 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 6 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
2 3 4 5 6	0.785714	0.714286	0.785714	0.747032	1.000000	1.000000	1.000000	1.000000
2 3 4 5 7	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
2 3 4 5 8	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
2 3 4 5 9	0.821429	0.767857	0.821429	0.793271	1.000000	1.000000	1.000000	1.000000
2 3 4 6 7	0.892857	0.839286	0.892857	0.864907	1.000000	1.000000	1.000000	1.000000
2 3 4 6 8	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
2 3 4 6 9	0.857143	0.821429	0.857143	0.838302	1.000000	1.000000	1.000000	1.000000
2 3 4 7 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
2 3 4 7 9	0.857143	0.821429	0.857143	0.838509	1.000000	1.000000	1.000000	1.000000
2 3 4 8 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
2 3 5 6 7	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
2 3 5 6 8	0.892857	0.839286	0.892857	0.863699	1.000000	1.000000	1.000000	1.000000
2 3 5 6 9	0.821429	0.767857	0.821429	0.792271	1.000000	1.000000	1.000000	1.000000
2 3 5 7 8	0.785714	0.714286	0.785714	0.748284	1.000000	1.000000	1.000000	1.000000

2 3 5 7 9	0.857143	0.821429	0.857143	0.838509	1.000000	1.000000	1.000000	1.000000
2 3 5 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
2 3 6 7 8	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
2 3 6 7 9	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
2 3 6 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
2 3 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
2 4 5 6 7	0.750000	0.732143	0.750000	0.740602	0.714286	0.642857	0.714286	0.675729
2 4 5 6 8	0.750000	0.666667	0.750000	0.704417	0.821429	0.744048	0.821429	0.780723
2 4 5 6 9	0.750000	0.660714	0.750000	0.700794	1.000000	1.000000	1.000000	1.000000
2 4 5 7 8	0.857143	0.797619	0.857143	0.824546	0.892857	0.839286	0.892857	0.863699
2 4 5 7 9	0.857143	0.821429	0.857143	0.838509	1.000000	1.000000	1.000000	1.000000
2 4 5 8 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
2 4 6 7 8	0.892857	0.851190	0.892857	0.870785	0.964286	0.946429	0.964286	0.954969
2 4 6 7 9	0.857143	0.821429	0.857143	0.838509	1.000000	1.000000	1.000000	1.000000
2 4 6 8 9	0.892857	0.839286	0.892857	0.864907	1.000000	1.000000	1.000000	1.000000
2 4 7 8 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
2 5 6 7 8	0.750000	0.708333	0.750000	0.727928	0.928571	0.892857	0.928571	0.909938
2 5 6 7 9	0.857143	0.821429	0.857143	0.838509	1.000000	1.000000	1.000000	1.000000
2 5 6 8 9	0.892857	0.839286	0.892857	0.863699	1.000000	1.000000	1.000000	1.000000
2 5 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
2 6 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
3 4 5 6 7	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
3 4 5 6 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
3 4 5 6 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
3 4 5 7 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3 4 5 7 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
3 4 5 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3 4 6 7 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3 4 6 7 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
3 4 6 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3 4 7 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3 5 6 7 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000

3 5 6 7 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
3 5 6 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3 5 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
3 6 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
4 5 6 7 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
4 5 6 7 9	0.821429	0.767857	0.821429	0.793478	1.000000	1.000000	1.000000	1.000000
4 5 6 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
4 5 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
4 6 7 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
5 6 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 3 4 5 6	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
1 2 3 4 5 7	0.857143	0.821429	0.857143	0.838509	1.000000	1.000000	1.000000	1.000000
1 2 3 4 5 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 2 3 4 5 9	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
1 2 3 4 6 7	0.892857	0.839286	0.892857	0.864907	1.000000	1.000000	1.000000	1.000000
1 2 3 4 6 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 2 3 4 6 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 2 3 4 7 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 3 4 7 9	0.857143	0.821429	0.857143	0.838509	1.000000	1.000000	1.000000	1.000000
1 2 3 4 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 2 3 5 6 7	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 3 5 6 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 3 5 6 9	0.928571	0.928571	0.928571	0.928571	1.000000	1.000000	1.000000	1.000000
1 2 3 5 7 8	0.785714	0.714286	0.785714	0.748284	1.000000	1.000000	1.000000	1.000000
1 2 3 5 7 9	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
1 2 3 5 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 3 6 7 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 3 6 7 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 2 3 6 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 3 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 4 5 6 7	0.857143	0.821429	0.857143	0.838509	0.928571	0.892857	0.928571	0.909938
1 2 4 5 6 8	0.928571	0.892857	0.928571	0.909938	0.928571	0.892857	0.928571	0.909938

1 2 4 5 6 9	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
1 2 4 5 7 8	0.964286	0.946429	0.964286	0.954969	0.964286	0.946429	0.964286	0.954969
1 2 4 5 7 9	0.857143	0.821429	0.857143	0.838509	1.000000	1.000000	1.000000	1.000000
1 2 4 5 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 4 6 7 8	0.964286	0.946429	0.964286	0.954969	0.964286	0.946429	0.964286	0.954969
1 2 4 6 7 9	0.857143	0.821429	0.857143	0.838509	1.000000	1.000000	1.000000	1.000000
1 2 4 6 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 2 4 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 5 6 7 8	0.964286	0.946429	0.964286	0.954969	0.964286	0.946429	0.964286	0.954969
1 2 5 6 7 9	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
1 2 5 6 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 5 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 6 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 4 5 6 7	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 3 4 5 6 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 3 4 5 6 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 4 5 7 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 3 4 5 7 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 3 4 5 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 3 4 6 7 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 3 4 6 7 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 4 6 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 3 4 7 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 3 5 6 7 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 5 6 7 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 3 5 6 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 3 5 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 6 7 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 4 5 6 7 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 4 5 6 7 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 4 5 6 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 4 5 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000

1 4 6 7 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 5 6 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
2 3 4 5 6 7	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
2 3 4 5 6 8	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
2 3 4 5 6 9	0.821429	0.767857	0.821429	0.793271	1.000000	1.000000	1.000000	1.000000
2 3 4 5 7 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
2 3 4 5 7 9	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
2 3 4 5 8 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
2 3 4 6 7 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
2 3 4 6 7 9	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
2 3 4 6 8 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
2 3 4 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
2 3 5 6 7 8	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
2 3 5 6 7 9	0.857143	0.821429	0.857143	0.838509	1.000000	1.000000	1.000000	1.000000
2 3 5 6 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
2 3 5 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
2 3 6 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
2 4 5 6 7 8	0.892857	0.851190	0.892857	0.870785	0.964286	0.946429	0.964286	0.954969
2 4 5 6 7 9	0.857143	0.821429	0.857143	0.838509	1.000000	1.000000	1.000000	1.000000
2 4 5 6 8 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
2 4 5 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
2 4 6 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
2 5 6 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
3 4 5 6 7 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3 4 5 6 7 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
3 4 5 6 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3 4 5 7 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3 4 6 7 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
3 5 6 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
4 5 6 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 3 4 5 6 7	0.857143	0.821429	0.857143	0.838509	1.000000	1.000000	1.000000	1.000000
1 2 3 4 5 6 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

1 2 3 4 5 6 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 3 4 5 7 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 3 4 5 7 9	0.857143	0.821429	0.857143	0.838509	1.000000	1.000000	1.000000	1.000000
1 2 3 4 5 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 2 3 4 6 7 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 3 4 6 7 9	0.892857	0.839286	0.892857	0.864907	1.000000	1.000000	1.000000	1.000000
1 2 3 4 6 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 2 3 4 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 3 5 6 7 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 3 5 6 7 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 2 3 5 6 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 3 5 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 3 6 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 4 5 6 7 8	0.964286	0.946429	0.964286	0.954969	0.964286	0.946429	0.964286	0.954969
1 2 4 5 6 7 9	0.857143	0.821429	0.857143	0.838509	0.964286	0.946429	0.964286	0.954969
1 2 4 5 6 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 2 4 5 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 4 6 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 5 6 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 4 5 6 7 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 3 4 5 6 7 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
1 3 4 5 6 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 3 4 5 7 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 3 4 6 7 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 3 5 6 7 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 4 5 6 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
2 3 4 5 6 7 8	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
2 3 4 5 6 7 9	0.892857	0.875000	0.892857	0.883540	1.000000	1.000000	1.000000	1.000000
2 3 4 5 6 8 9	0.928571	0.892857	0.928571	0.909938	1.000000	1.000000	1.000000	1.000000
2 3 4 5 7 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
2 3 4 6 7 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
2 3 5 6 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000

2 4 5 6 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
3 4 5 6 7 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 2 3 4 5 6 7 8	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 3 4 5 6 7 9	0.892857	0.839286	0.892857	0.864907	1.000000	1.000000	1.000000	1.000000
1 2 3 4 5 6 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 2 3 4 5 7 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 2 3 4 6 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 3 5 6 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 2 4 5 6 7 8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
1 3 4 5 6 7 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
2 3 4 5 6 7 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 2 3 4 5 6 7 8 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Table S10: SVM and LDA data for 1- and 2-feature combinations including CD43 (feature 9). Combinations showing 100% accuracy are highlighted.

	SVM				LDA			
Feature set	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
1 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
2 9	0.535714	0.410714	0.535714	0.463946	1.000000	1.000000	1.000000	1.000000
3 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
4 9	0.821429	0.732143	0.821429	0.773637	1.000000	1.000000	1.000000	1.000000
5 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000
6 9	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
7 9	0.892857	0.857143	0.892857	0.874142	1.000000	1.000000	1.000000	1.000000
8 9	0.964286	0.946429	0.964286	0.954969	1.000000	1.000000	1.000000	1.000000

4 SPSS-based LDA

Table S11: LDA classification results using CD43 only

Classification Results[a,c]											
Items		pH	Predicted Group Membership								Total
			2.8	3.9	4.9	6.6	6.9	8	9.1	10	
Original	Count	2.76	4	0	0	0	0	0	0	0	4
		3.92	0	4	0	0	0	0	0	0	4
		4.87	0	0	4	0	0	0	0	0	4
		6.64	0	0	0	4	0	0	0	0	4
		6.9	0	0	0	0	4	0	0	0	4
		8	0	0	0	0	0	4	0	0	4
		9.05	0	0	0	0	0	0	4	0	4
		10	0	0	0	0	0	0	0	4	4
	%	2.76	100	0	0	0	0	0	0	0	100
		3.92	0	100	0	0	0	0	0	0	100
		4.87	0	0	100	0	0	0	0	0	100
		6.64	0	0	0	100	0	0	0	0	100
		6.9	0	0	0	0	100	0	0	0	100
		8	0	0	0	0	0	100	0	0	100
		9.05	0	0	0	0	0	0	100	0	100
		10	0	0	0	0	0	0	0	100	100
Cross-validated[b]	Count	2.76	4	0	0	0	0	0	0	0	4
		3.92	0	4	0	0	0	0	0	0	4
		4.87	0	0	4	0	0	0	0	0	4
		6.64	0	0	0	4	0	0	0	0	4
		6.9	0	0	0	0	4	0	0	0	4
		8	0	0	0	0	0	4	0	0	4
		9.05	0	0	0	0	0	0	4	0	4
		10	0	0	0	0	0	0	0	4	4
	%	2.76	100	0	0	0	0	0	0	0	100
		3.92	0	100	0	0	0	0	0	0	100
		4.87	0	0	100	0	0	0	0	0	100
		6.64	0	0	0	100	0	0	0	0	100
		6.9	0	0	0	0	100	0	0	0	100
		8	0	0	0	0	0	100	0	0	100
		9.05	0	0	0	0	0	0	100	0	100
		10	0	0	0	0	0	0	0	100	100
a. 100.0% of original grouped cases correctly classified.											
b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.											
c. 100.0% of cross-validated grouped cases correctly classified.											

Table S1: Summary of canonical discriminant function by using CD43 only

Eigenvalues				
Function	Eigenvalue	% of Variance	Cumulative %	Canonical Correlation
1	2866.052	100	100	1
Wilks' Lambda				
Test of Function(s)	Wilks' Lambda	Chi-square	df	Sig.
1	0	210.968	7	<.001

Table S2: Functions at group centroids using CD43 only

pH level	Function
2.76	-2.288
3.92	44.289
4.87	69.607
6.64	21.624
6.9	28.512
8	-38.346
9.05	-58.543
10.03	-64.855

Table S34: LDA classification of array built by CD43 and CD69.

Classification Results[a,c]											
Items		pH	Predicted Group Membership								Total
			2.8	3.9	4.9	6.6	6.9	8	9.1	10	
Original	Count	2.76	4	0	0	0	0	0	0	0	4
		3.92	0	4	0	0	0	0	0	0	4
		4.87	0	0	4	0	0	0	0	0	4
		6.64	0	0	0	4	0	0	0	0	4
		6.9	0	0	0	0	4	0	0	0	4
		8	0	0	0	0	0	4	0	0	4
		9.05	0	0	0	0	0	0	4	0	4
		10	0	0	0	0	0	0	0	4	4
	%	2.76	100	0	0	0	0	0	0	0	100
		3.92	0	100	0	0	0	0	0	0	100
		4.87	0	0	100	0	0	0	0	0	100
		6.64	0	0	0	100	0	0	0	0	100
		6.9	0	0	0	0	100	0	0	0	100
		8	0	0	0	0	0	100	0	0	100
		9.05	0	0	0	0	0	0	100	0	100
		10	0	0	0	0	0	0	0	100	100
Cross-validated[b]	Count	2.76	4	0	0	0	0	0	0	0	4
		3.92	0	4	0	0	0	0	0	0	4
		4.87	0	0	4	0	0	0	0	0	4
		6.64	0	0	0	4	0	0	0	0	4
		6.9	0	0	0	0	4	0	0	0	4
		8	0	0	0	0	0	4	0	0	4
		9.05	0	0	0	0	0	0	4	0	4
		10	0	0	0	0	0	0	0	4	4
	%	2.76	100	0	0	0	0	0	0	0	100
		3.92	0	100	0	0	0	0	0	0	100
		4.87	0	0	100	0	0	0	0	0	100
		6.64	0	0	0	100	0	0	0	0	100
		6.9	0	0	0	0	100	0	0	0	100
		8	0	0	0	0	0	100	0	0	100
		9.05	0	0	0	0	0	0	100	0	100
		10	0	0	0	0	0	0	0	100	100

a. 100.0% of original grouped cases correctly classified.

b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.

c. 100.0% of cross-validated grouped cases correctly classified.

Table S4: Summary of canonical discriminant function using array built by CD43 and CD69.

Eigenvalues				
Function	Eigenvalue	% of Variance	Cumulative %	Canonical Correlation
1	2867.290[a]	84.3	84.3	1
2	534.396[a]	15.7	100	0.999
a. First 2 canonical discriminant functions were used in the analysis.				
Wilks' Lambda				
Test of Function(s)	Wilks' Lambda	Chi-square	df	Sig.
1 through 2	0	370.356	14	<.001
2	0.002	163.358	6	<.001

Table S5: Functions at group centroids using array built by CD43 and CD69

pH level	Function	
	1	2
2.76	-3.252	-41.79
3.92	43.839	-20.034
4.87	69.764	6.006
6.64	22.226	25.888
6.9	28.856	14.576
8	-38.091	11.478
9.05	-58.501	2.504
10.03	-64.841	1.372

Table S6: LDA classification results by using array built by CD43 and CD25.

Classification Results[a,c]											
Items		pH	Predicted Group Membership								Total
			2.8	3.9	4.9	6.6	6.9	8	9.1	10	
Original	Count	2.76	4	0	0	0	0	0	0	0	4
		3.92	0	4	0	0	0	0	0	0	4
		4.87	0	0	4	0	0	0	0	0	4
		6.64	0	0	0	4	0	0	0	0	4
		6.9	0	0	0	0	4	0	0	0	4
		8	0	0	0	0	0	4	0	0	4
		9.05	0	0	0	0	0	0	4	0	4
		10	0	0	0	0	0	0	0	4	4
	%	2.76	100	0	0	0	0	0	0	0	100
		3.92	0	100	0	0	0	0	0	0	100
		4.87	0	0	100	0	0	0	0	0	100
		6.64	0	0	0	100	0	0	0	0	100
		6.9	0	0	0	0	100	0	0	0	100
		8	0	0	0	0	0	100	0	0	100
		9.05	0	0	0	0	0	0	100	0	100
		10	0	0	0	0	0	0	0	100	100
Cross-validated[b]	Count	2.76	4	0	0	0	0	0	0	0	4
		3.92	0	4	0	0	0	0	0	0	4
		4.87	0	0	4	0	0	0	0	0	4
		6.64	0	0	0	4	0	0	0	0	4
		6.9	0	0	0	0	4	0	0	0	4
		8	0	0	0	0	0	4	0	0	4
		9.05	0	0	0	0	0	0	4	0	4
		10	0	0	0	0	0	0	0	4	4
	%	2.76	100	0	0	0	0	0	0	0	100
		3.92	0	100	0	0	0	0	0	0	100
		4.87	0	0	100	0	0	0	0	0	100
		6.64	0	0	0	100	0	0	0	0	100
		6.9	0	0	0	0	100	0	0	0	100
		8	0	0	0	0	0	100	0	0	100
		9.05	0	0	0	0	0	0	100	0	100
		10	0	0	0	0	0	0	0	100	100
a. 100.0% of original grouped cases correctly classified.											
b. Cross validation is done only for those cases in the analysis. In cross validation, each case is classified by the functions derived from all cases other than that case.											
c. 100.0% of cross-validated grouped cases correctly classified.											

Table S7: Summary of canonical discriminant function using array built by CD43 and CD25.

Eigenvalues				
Function	Eigenvalue	% of Variance	Cumulative %	Canonical Correlation
1	3357.485[a]	98.2	98.2	1
2	61.584[a]	1.8	100	0.992
a. First 2 canonical discriminant functions were used in the analysis.				
Wilks' Lambda				
Test of Function(s)	Wilks' Lambda	Chi-square	df	Sig.
1 through 2	0	318.65	14	<.001
2	0.016	107.549	6	<.001

Table S89: Functions at group centroids by using array built by CD43 and CD25.

pH level	Function	
	1	2
2.76	-8.219	13.707
3.92	45.033	7.119
4.87	75.528	-0.162
6.64	26.668	-7.706
6.9	34.036	-7.47
8	-40.047	-3.638
9.05	-62.812	-1.562
10.03	-70.187	-0.288

5 MATLAB based GPR

5.1 MATLAB GPR code development

For developing a pH sensing array for GPR prediction application, we developed the GPR based MATLAB. GPR is a non-parametric, Bayesian machine learning method that can be used for regression problems. It is based on the Gaussian Process, a collection of random variables where any finite subset of these variables follows a multivariate Gaussian distribution. GPR is particularly useful in cases where the relationship between the input features and the target variable is complex and non-linear. In GPR, the objective is to learn a function that best maps the input features to the target variable. This function is modelled as a Gaussian Process with a mean function and a covariance function (also called the kernel function). The kernel function measures the similarity between the input data points and encodes the correlation between the target values. The choice of the kernel function can have a significant impact on the performance of the model. Here introduces the MATLAB code development based on the code flow. A step-by-step explanation is provided in supporting material. The script performed data preprocessing, model training, prediction, and performance evaluation using various feature combinations.

Data Loading and Preprocessing: The code began by clearing the workspace, closing all figures, and turning off warnings. It then loaded the training and test datasets from Excel files:

```
clc
clear all
close all
warning off

%% Load data
train_data = xlsread('DataHaobo.xlsx');
test_data_full = xlsread('Test.xlsx');
```

The features (`X_train` and `X_test`) and target variables (`y_train` and `y_test_actual`) were extracted from the datasets. The target variable for training was also converted to a logarithmic scale to improve the performance of the GPR model:

```
% Features and target variables
X_train = train_data(:, 2:end); % Features for training
y_train = train_data(:, 1); % Target for training

X_test = test_data_full(:, 2:end); % Features for testing
y_test_actual = test_data_full(:, 1); % Actual pH for testing

% Convert target to logarithmic scale for GPR
y_train_log = log(y_train);
```

Feature Combinations and Directory Setup: The script generated all possible combinations of features to be used for model training and initialised directories for storing the results:

```
% Initialize combinations and results storage
num_features = size(X_train, 2);
combinations = cell(num_features, 1);

    combinations{k} = nchoosek(1:num_features, k);
end

% Create directories for results if they don't exist
if ~exist('GPR_result', 'dir')
    mkdir('GPR_result')
end
if ~exist('GPR_result/Figures', 'dir')
    mkdir('GPR_result/Figures')
end
if ~exist('GPR_result/EditableFigures', 'dir')
    mkdir('GPR_result/EditableFigures')
end
```

Model Training and Prediction: For each combination of features, the script trained a GPR model and made predictions on the test data. The Kernel Function with SquaredExponential was applied during the GPR trainer modelling. There were 4 test set for each of the array selection. It then computed various evaluation metrics and saved the results:

```
% Loop through all feature combinations
for k = 1:num_features
    for i = 1:size(combinations{k}, 1)
        feature_combination = combinations{k}(i, :);
        X_train_comb = X_train(:, feature_combination);
        X_test_comb = X_test(:, feature_combination);

        % Train the GPR model
        gprMdl = fitrgp(X_train_comb, y_train_log);

        % Predict target variable for test data
        y_pred_log = predict(gprMdl, X_test_comb);
        y_pred = exp(y_pred_log);

        % Compute predictions for all training data points
        [ypred_log, ~, yci_log] = predict(gprMdl, X_train_comb);
        ypred = exp(ypred_log);
        yci = exp(yci_log);

        % Calculate average predicted pH value and standard deviation for n=4
        mean_y_pred = mean(y_pred);
        std_y_pred = std(y_pred);
        prediction_range = [mean_y_pred - std_y_pred, mean_y_pred + std_y_pred];

        % Convert kernel information to string
        kernelInfoStr = struct2str(gprMdl.KernelInformation);

        % Save the results in a text file
```

```
    result_filename = ['GPR_result/FeatureCombination_', mat2str(feature_combination), '.txt'];
    fileID = fopen(result_filename, 'w');

    fprintf(fileID, 'Feature Combination: %s\n', mat2str(feature_combination));
    fprintf(fileID, 'Kernel Information: %s\n', kernelInfoStr);
    fprintf(fileID, 'Kernel Function: %s\n', gprMdl.KernelFunction);
    fprintf(fileID, 'Standard deviation of predicted values (n=4): %.10f\n', std_y_pred);
    fprintf(fileID, 'Test pH of %.2f gave an average prediction of %.10f ± %.10f (standard deviation, n=%d)\n',
y_test_actual(1), mean_y_pred, std_y_pred, size(X_test_comb, 1));
    fprintf(fileID, 'Prediction range: %.10f to %.10f\n', prediction_range(1), prediction_range(2));

    % Calculate and display the coefficient of determination (R²), MSE, and MAE
    R2 = 1 - sum((y_train - ypred).^2) / sum((y_train - mean(y_train)).^2);
    MSE = mean((y_train - ypred).^2);
    MAE = mean(abs(y_train - ypred));

    fprintf(fileID, 'Coefficient of Determination (R²): %.10f\n', R2);
    fprintf(fileID, 'Mean Squared Error (MSE): %.10f\n', MSE);
    fprintf(fileID, 'Mean Absolute Error (MAE): %.10f\n', MAE);

    fclose(fileID);

    % Append results to the result table
    result_table = [result_table; {mat2str(feature_combination), kernelInfoStr, gprMdl.KernelFunction,
std_y_pred, mean_y_pred, std_y_pred, size(X_test_comb, 1), prediction_range(1), prediction_range(2), R2,
MSE, MAE}];
```

Visualisation and Results Saving: The code visualised the predicted versus actual pH values for each feature combination and saved the figures in both TIFF and MATLAB editable formats. It also saved the results to an Excel file for further analysis:

```
    % Plot predicted vs actual values
    figure;
    hold on;
    % Add error bars to the training data
    error_bar = (yci(:, 2) - yci(:, 1)) / 2;
    errorbar(y_train, ypred, error_bar, 'bo');
    plot(y_train, y_train, 'k-');
    xlabel('Actual pH');
    ylabel('Predicted pH');
    title(['GPR - Feature Combination: ', mat2str(feature_combination)]);
    legend('Predicted vs actual', 'Ideal line', 'Location', 'Northwest');
    % Plot predicted test points
    for j = 1:size(X_test_comb, 1)
      plot(y_test_actual(j), y_pred(j), 'r*', 'MarkerSize', 10);
    end

    % Save figure in TIFF format
    print('-dtiff', '-r1200', ['GPR_result/Figures/GPR_FeatureCombination_', mat2str(feature_combination),
'.tiff']);

    % Save figure in MATLAB editable format
```

```
        saveas(gcf, ['GPR_result/EditableFigures/GPR_FeatureCombination_', mat2str(feature_combination),
'.fig']);

        % Close the plot window
        close(gcf);
    end
end

% Save results to Excel file
result_headers = {'Feature Combination', 'Kernel Information', 'Kernel Function', 'Std Deviation', 'Mean
Prediction', 'Std Dev', 'Num Test Points', 'Prediction Range Min', 'Prediction Range Max', 'R²', 'MSE', 'MAE'};
result_table = cell2table(result_table, 'VariableNames', result_headers);
writetable(result_table, 'GPR_result/GPR_results.xlsx');
```

Utility Functions: The script included utility functions for converting structures and arrays to strings, which were used in the process of saving the results:

```
% Function to convert structure to string
function str = struct2str(structVar)
    str = '';
    fields = fieldnames(structVar);
    for i = 1:numel(fields)
        fieldName = fields{i};
        fieldValue = structVar.(fieldName);
        if isstruct(fieldValue)
            fieldValueStr = struct2str(fieldValue);
        else
            fieldValueStr = array2str(fieldValue);
        end
        str = [str, fieldName, ': ', fieldValueStr, '; '];
    end
end

% Function to convert array to string
function str = array2str(array)
    if isnumeric(array) || islogical(array)
        str = mat2str(array);
    elseif ischar(array)
        str = array;
    elseif isstring(array)
        str = char(array);
    else
        str = 'Unsupported data type';
    end
end

% Function to extract and display kernel parameters names and values
function getKernelParameters(gprMdl)
    kernelInfo = gprMdl.KernelInformation;
    paramName = gprMdl.KernelInformation.KernelParameterNames;
    paramValues = kernelInfo.KernelParameters;
```

```
    disp('Kernel parameter names:');
    disp(paramName);

    disp('Kernel parameter values:');
    for i = 1:numel(paramValues)
        fprintf('%.10f\n', paramValues(i));
    end
end
```

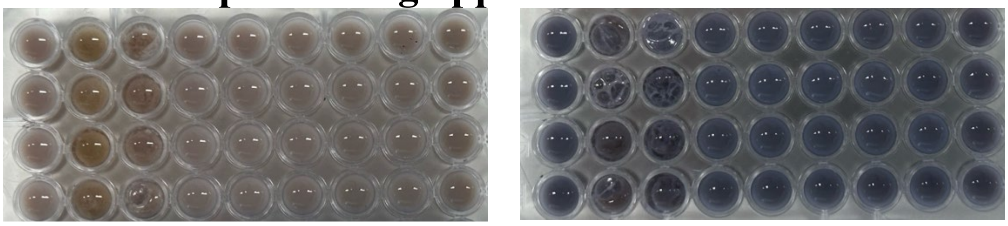
In summary, this paper presents a well-structured, modular, and efficient MATLAB code implementation of Gaussian Process Regression (GPR) for predicting pH values. The code adheres to design principles, ensuring an organized and systematic approach to addressing the regression problem. This comprehensive approach ensured robust evaluation of different feature combinations, providing insights into the optimal configuration for accurate pH prediction using GPR. Modularity is a key feature of the code, with distinct sections dedicated to data loading, pre-processing, model training, prediction, statistical analysis, and visualization. This separation not only improves readability and maintainability but also makes it more user-friendly. The code's flexibility allows for easy adaptation to other datasets or regression problems, requiring only minor adjustments to the input data file and feature/target variable assignments. One of the code's main strengths is its user-friendliness, enabling users with limited coding and MATLAB experience to quickly adapt it for specific regression problems with minimal modifications. Overall, the MATLAB code described in this paper offers a practical, organized, and user-friendly solution for implementing GPR, making it a valuable tool for a wide range of applications and users.

5.2 GPR results

Table S20: The most promising feature combination is highlighted associate with their GPR performance

Feature Combination	Test pH	Kernel Parameters	Std Deviation	Mean Prediction	Num Test Points	Prediction Range Min	Prediction Range Max	R^2	MSE	MAE
9	6.9	84.75922314882;0.362353037454844	0.120	7.259	4	7.139	7.379	1.000	0.000	0.009
3 9	6.9	427.6691970988;0.316333780807546	0.110	6.795	4	6.685	6.904	1.000	0.000	0.009
6 9	6.9	13068.578066183;0.346680829348127	0.069	7.154	4	7.085	7.223	1.000	0.002	0.024
3 9	4.07	86829.8011803673;0.468063851297936	0.106	4.039	4	3.934	4.145	0.987	0.055	0.165
3 9	9.05	75299.2182486293;0.437275578262553	0.035	8.970	4	8.934	9.006	0.996	0.019	0.102

6 GPR based pH sensing application with milk



CD69 CD43

Figure S12: Naked eye visual observation of CDs sensor array applied with milk under varying pH conditions (Each photo contains four replicates, with each row representing one replicate. From left to right, the columns represent CDs in MilliQ water and spoiled milk, followed by CDs with spoiled milk exposed to pH levels 3.06, 4.04, 5.55, 6.15, 7.2, 8.05, 9.25, and 10.09).