

---

## Table of Contents

% Summary: .....	1
% Preprocessing required: .....	1
% Brief Overview of pre-processing: .....	2
% ADDITIONAL processing/files: .....	2
% Reference: .....	2
% Standard figure parameters .....	3
% Flags and initializations .....	3
% Load/Read twin/sibling pair info. ....	3
% Load subject proximity graph: .....	4
% Create Modality combinations (Hard coded for illustration purpose) .....	5
%%%%%%%%% .....	5
% Example 1: Compact fingerprint generation .....	5
%%%%%%%%% .....	5
%%%%%%%%% .....	6
% Example 2: Compact fingerprint analysis: d-prime and -log10 p-value MZ vs DZ plots (Fig 2.) .....	6
%%%%%%%%% .....	6
% Plot absolute d-prime and p-val for increasing fingerprint length/size .....	7
% Plot -log10 p-value (ttest2): MZ vs DZ for increasing number of eig vecs .....	8
%%%%%%%%% .....	9
% Example 3: Compact fingerprint comparison plots: MZ, DZ, FS pairs (Fig 3.) .....	9
%%%%%%%%% .....	9
%%%%%%%%% .....	13
% Example 4: Rank Reterival Analysis: Mean Average Precision Table (Table 2, partial) .....	13
%%%%%%%%% .....	13
% Display mean average precision (MAP) values .....	15
%%%%%%%%% .....	15
% Example 5: Mean Recall@10 values (Supplement material Fig Table 8, partial) .....	15
%%%%%%%%% .....	15
%%%%%%%%% .....	16
% Example 6: Mean Recall@k Plots (Supplement material Fig 1.) .....	16
%%%%%%%%% .....	16
%%%%%%%%% .....	19
% Example 7: Relative informativeness Table: MZ, DZ, FS (Table 3) .....	19
%%%%%%%%% .....	19
% Display realtive identification percentages: MZ, DZ, FS tables .....	20
%%%%%%%%% .....	21
%%%%%%%%% .....	21

## % Summary:

1. MATLAB demo function
  - generate sample plots reported in paper

## % Preprocessing required:

\* REQUIREMENTS:

- 
1. report.txt files for each modality (see below)
  2. twin/sibling pair ids (numeric id's between 1 to number of subjects/images based on the subject ordering/reading list) for example: MZ sub id set and MZ pair id set etc..
  3. Note: The randomized sample includes pairs of MZ; DZ; Full-Siblings (FS); Maternal Half Siblings (MHS) and Paternal Half Siblings (PHS)

NOTE: a sample set of report files and twin/sibling pair ids (randomized) have been included in data folder for illustration purposes

## % Brief Overview of pre-processing:

1. We assume that Bag-of-features (BoF) have been generated and approximate matching has been performed using the code available at <http://www.matthewtoews.com/>  
THE PRE-PROCESSING involves running:
  - i. featExtract.exe for each image (.nii files): output will be .key files containing 3D SIFT features
  - ii. featMatchMultiple -t 2 -r- -n 20 \*.key (for all .key files of a given modality): output will be multiple files including report.txt
2. report.txt generated for each modality will act as a input for the following analysis
3. We have included sample report files (Random permutation applied on our set)
4. Note: cross check the file order read in featMatchMultiple.exe and the sibling/twin pair IDs

## % ADDITIONAL processing/files:

1. feature match visualization requires 3D SIFT feature files (not included here)
2. Significance values (-log10 pvalues) in paper are based on sample size matches using bipartite matching (age-based); However, sample illustration uses random sample size matching
3. Similarly hemisphere asymmetry analysis and pairwise feature correspondence analysis requires restricted data and thus have been excluded here

## % Reference:

Multi-modal brain fingerprinting: a manifold approximation based  
Authors:  
Kuldeep Kumar (kkumar@livia.etsmtl.ca),  
Laurent Chauvin  
Matthew Toews (Matthew.Toews@etsmtl.ca)  
Olivier Colliot and  
Christian Desrosiers (christian.desrosiers@etsmtl.ca)

LIVIA, ETS Montreal, Canada January 2018

---

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

## % Standard figure parameters

```
Line_width = 3 ;
Marker_size= 14;
leg_FontSize=13;
gca_FontSize = 32;

color_option{1} = 'r' ;
color_option{2} = 'g' ;
color_option{3} = 'b' ;
color_option{4} = 'c' ;
color_option{5} = 'm' ;
color_option{6} = 'y' ;
color_option{7} = 'k' ;

marker_option{1} = '*' ;
marker_option{2} = 'o' ;
marker_option{3} = 'd' ;
marker_option{4} = 's' ;
marker_option{5} = 'x' ;
marker_option{6} = '^' ;
marker_option{7} = '+' ;
```

## % Flags and initializations

```
flag to display figure: 1: yes, 0: no

flag_display_figure =1;

% labels for sibling types
cell_sibling_type{1} = 'MZ';
cell_sibling_type{2} = 'DZ';
cell_sibling_type{3} = 'FS';
cell_sibling_type{4} = 'MHS';
cell_sibling_type{5} = 'FHS';

% Define the twin/sibling types to be used for analyses
max_sib_type = 3;
% 3: MZ, DZ, FS; 5: MZ, DZ, FS, MHS and FHS also
```

## % Load/Read twin/sibling pair info.

```
%load SAMPLE twin/sibling subject id and pair id information
twin_sib_info_file = '../data/SAMPLE_cell_subID_and_sibpairID.mat';
load(twin_sib_info_file);
% output will be variable: 'SAMPLE_cell_subID_and_sibpairID'

% variable: a cell array of size 5 x 2

% dim 1 correspond to twin/sibling type:
```

---

```

%      1. MZ; 2. DZ; 3. FS; 4. MHS; and 5. PHS

% dim 2: corresponds to sub ids and pair ids

%      1. sub_id_set  2. pair_id_set

% NOTE: the SAMPLE info is randomized before sharing

```

## % Load subject proximity graph:

```

max_modality_combinations= 9;
% hard coded for illustration
cell_NxN_data_matrix = cell(max_modality_combinations,1);

% Set of modalities used for illustration
cell_modality{1} = 'FA';
cell_modality{2} = 'T1w 125mm';
cell_modality{3} = 'T2w 125mm';
cell_modality{4} = 'rfMRI';
cell_modality{5} = 'T1w+FA';
cell_modality{6} = 'T1w+rfMRI';
cell_modality{7} = 'FA+rfMRI';
cell_modality{8} = 'T1w+T2w+FA';
cell_modality{9} = 'T1w+T2w+FA+rfMRI';

%DATA: load SAMPLE report files (feature match count) for each
modality
cell_report_filename = { 'SAMPLE_report_FA.mat';
                        'SAMPLE_report_T1w_125mm';
                        'SAMPLE_report_T2w_125mm';

                        'SAMPLE_rfMRI_pairwise_pearson_corr_ICA100' };

% For report.txt files:
% 1. read feature match count files (report.txt)
% 2. Obtain Normalized Jaccard Similarity

cell_report_files = cell(3,1);

for loop_i=1:(length(cell_report_filename)-1)
    load(['../data/' cell_report_filename{loop_i}]);
    cell_report_files{loop_i,1} = SAMPLE_report_mat ;
    cell_NxN_data_matrix{loop_i} =
        compute_normalized_jaccard_similarity_matrix(cell_report_files(loop_i,1));
end

% read rfMRI NxN_data_matrix
load(['../data/'
    cell_report_filename{length(cell_report_filename)}]);
cell_NxN_data_matrix{length(cell_report_filename)} =
    SAMPLE_rfMRI_pairwise_pearson_corr_ICA100 ;

```

---

## % Create Modality combinations (Hard coded for illustration purpose)

```
% 1. T1w + FA
loop_i = 5;
cell_NxN_data_matrix{loop_i} =
compute_normalized_jaccard_similarity_matrix(cell_report_files(1:2));

% 2. T1w + rfMRI: (linear combination, with combination weights
computed using a grid
% search, to optimize MAP values )
loop_i = 6;
lambda = 0.2;
cell_NxN_data_matrix{loop_i} = lambda*cell_NxN_data_matrix{4}+(1-
lambda)*cell_NxN_data_matrix{2} ;

% 3. FA + rfMRI (linear combination, with combination weights
computed using a grid
% search, to optimize MAP values )
loop_i = 7;
lambda = 0.4;
cell_NxN_data_matrix{loop_i} = lambda*cell_NxN_data_matrix{4}+(1-
lambda)*cell_NxN_data_matrix{2} ;

% 4. T1w + T2w + FA (1.25mm)
loop_i=8;
cell_NxN_data_matrix{loop_i} =
compute_normalized_jaccard_similarity_matrix(cell_report_files(1:3));

% 5. T1w + T2w + FA + rfMRI
loop_i = 9;
lambda = 0.2;
cell_NxN_data_matrix{loop_i} = lambda*cell_NxN_data_matrix{4}+(1-
lambda)*cell_NxN_data_matrix{8} ;
```

%%  
%%

## % Example 1: Compact fingerprint generation

%%  
%%

```
array_modality_set = [ 1; 2; 5];
% FA, T1w and T1w + FA
num_spect_component = 150;
% number of eigen vectors in fingerprint
cell_compact_fingerprint = cell(length(array_modality_set),1);
```

---

```

for loop_i=1:length(array_modality_set)

    % compact fingerprint generation
    NxN_data_matrix =
    cell_NxN_data_matrix{array_modality_set(loop_i),1};
    cell_compact_fingerprint{loop_i,1} =
    compute_compact_fingerprint(NxN_data_matrix,num_spect_component);

end

```

%%%%%%%%%

%%%%%%%%%

## % Example 2: Compact fingerprint analysis: d-prime and -log10 p-value MZ vs DZ plots (Fig 2.)

%%%%%%%%%

%%%%%%%%%

```

%
% compute d-prime and -log10 p-value for pairwise fingerprint
% distance: MZ vs DZ
%

array_num_spect_component = (5:5:500)';
array_modality_set = [ 1; 2; 5];
    % FA, T1w and T1w + FA (all at 1.25mm resolution)

num_sibtype_comparisons = (max_sib_type*(max_sib_type-1))/2;
    % 1. MZ vs DZ, 2: MZ vs FS, 3: DZ vs FS

cell_d_prime =
cell(length(array_num_spect_component),num_sibtype_comparisons,length(array_modal
cell_pval_ttest2 =
cell(length(array_num_spect_component),num_sibtype_comparisons,length(array_modal
cell_pval_ranksum =
cell(length(array_num_spect_component),num_sibtype_comparisons,length(array_modal

for loop_i=1:length(array_modality_set)

    % compact fingerprint generation
    NxN_data_matrix =
    cell_NxN_data_matrix{array_modality_set(loop_i),1};
    temp_cell_compact_fingerprint =
    compute_compact_fingerprint_size_impact(NxN_data_matrix,array_num_spect_component

```

---

```

    for loop_num_spect = 1:length(array_num_spect_component)

        % compute Euclidean Distance between fingerprint pairs for MZ,
        DZ, FS
        mat_compact_fingerprint =
        temp_cell_compact_fingerprint{loop_num_spect,1};
        cell_pairwise_distance = cell(max_sib_type,1);

        for loop_sib_type=1:max_sib_type
            sub_id_set =
            SAMPLE_cell_subID_and_sibpairID{loop_sib_type,1};
            pair_id_set =
            SAMPLE_cell_subID_and_sibpairID{loop_sib_type,2};
            [cell_pairwise_distance{loop_sib_type,1}] =
            compute_pairwise_fingerprint_distance_siblings(sub_id_set,pair_id_set,mat_co
        end

        % compute d-prime and pval for MZ vs DZ, MZ vs FS, and DZ vs FS
        temp_count_loop_comparison =1;
        for loop_sib_type_1 = 1:(max_sib_type-1)
            array_pairwise_distance_1 =
            cell_pairwise_distance{loop_sib_type_1,1} ;

            for loop_sib_type_2 = (loop_sib_type_1 + 1):max_sib_type
                array_pairwise_distance_2 =
                cell_pairwise_distance{loop_sib_type_2,1} ;

            cell_d_prime{loop_num_spect,temp_count_loop_comparison,loop_i} =
            compute_d_prime(array_pairwise_distance_1,array_pairwise_distance_2);

            [cell_pval_ttest2{loop_num_spect,temp_count_loop_comparison,loop_i},cell_pval_ran
            =
            compute_pval_ttest2_ranksum(array_pairwise_distance_1,array_pairwise_distance_2);
            temp_count_loop_comparison = temp_count_loop_comparison
            + 1;
        end
    end
end
end

```

## % Plot absolute d-prime and p-val for increasing fingerprint length/size

Absolute d-prime: MZ vs DZ

```

temp_count_loop_comparison=1;
    % 1: MZ vs DZ, 2: MZ vs FS, .....
temp_data = cell_d_prime(:,temp_count_loop_comparison,1:
(length(array_modality_set)));

```

---

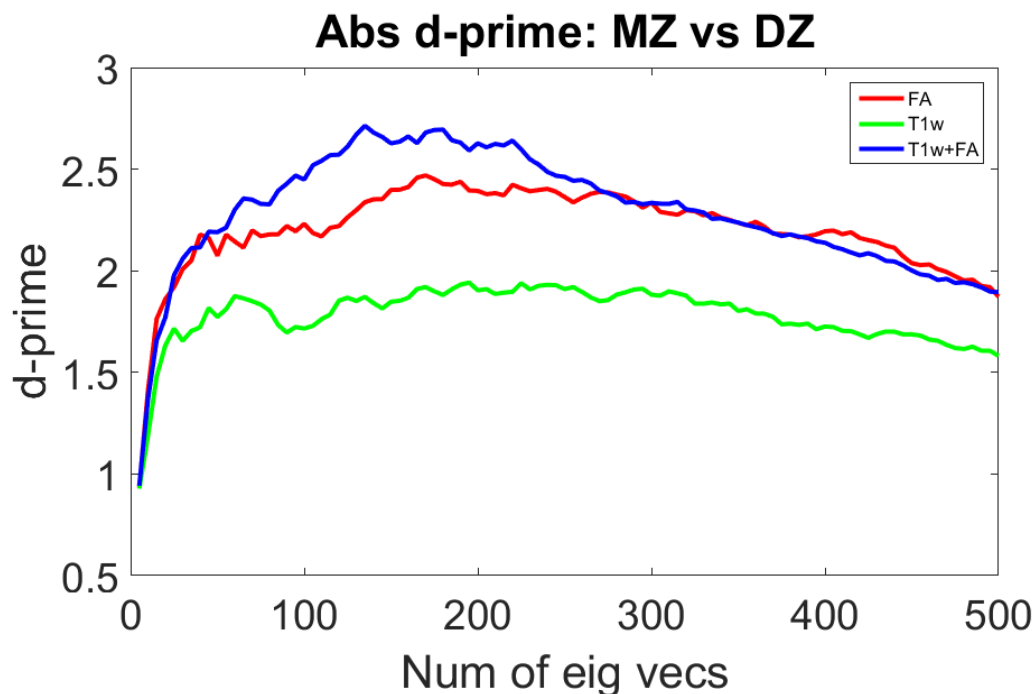
```

data_mat =
    abs((reshape(cell2mat(temp_data),size(temp_data,1),size(temp_data,3))));

figure_title = 'Abs d-prime: MZ vs DZ';
x_tick_array = [ 0; (100:100:500)'];
cell_label{1} = 'Num of eig vecs';
cell_label{2} = 'd-prime';
leg_location = 'northeast';
cell_legend = {'FA'; 'T1w'; 'T1w+FA'};
save_plot_name = [];
    % assign filename to save it else empty

plot_compact_fingerprint_analysis(data_mat,x_tick_array,figure_title,cell_legend,1

```



## % Plot -log10 p-value (ttest2): MZ vs DZ for increasing number of eig vecs

```

temp_count_loop_comparison=1;
    % 1: MZ vs DZ, 2: MZ vs FS, .....
temp_data = cell_pval_ttest2(:,temp_count_loop_comparison,1:
    (length(array_modality_set)));
data_mat =
    -1*log10((reshape(cell2mat(temp_data),size(temp_data,1),size(temp_data,3))));

figure_title = '-log10 p-value: MZ vs DZ';
x_tick_array = [ 0; (100:100:500)'];
cell_label{1} = 'Num of eig vecs';
cell_label{2} = '-log10 pvalue';
leg_location = 'northeast';

```

---



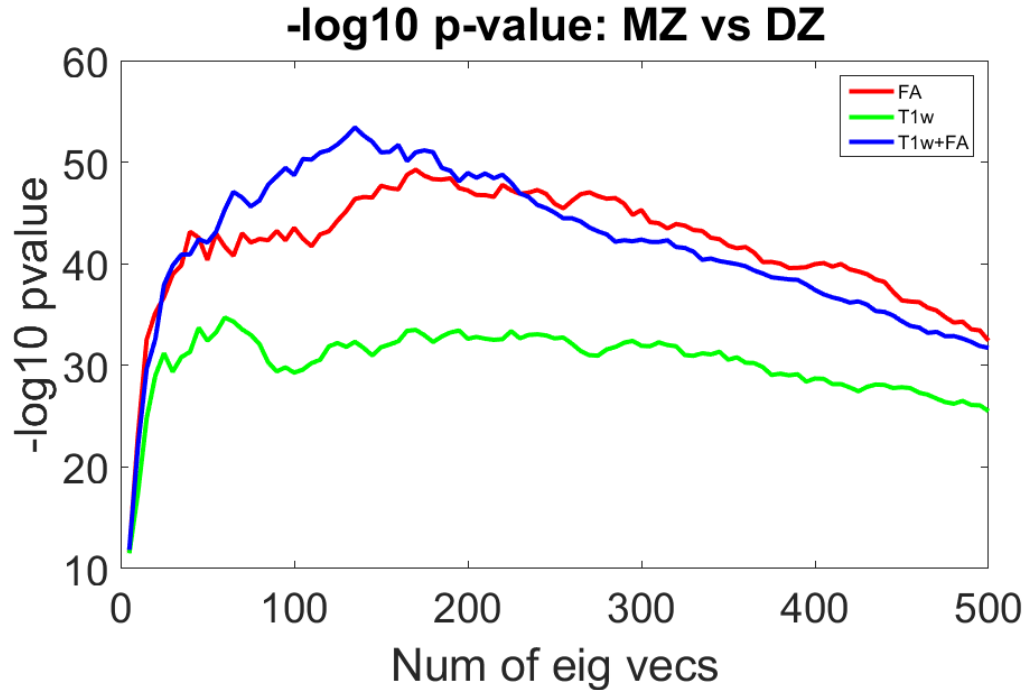
---

```

cell_legend = {'FA'; 'T1w'; 'T1w+FA'};
save_plot_name = [];
    % assign filename to save it else empty

plot_compact_fingerprint_analysis(data_mat,x_tick_array,figure_title,cell_legend,1

```



```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

**% Example 3: Compact fingerprint comparison plots: MZ, DZ, FS pairs (Fig 3.)**

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

array_modality_set = [ 1; 2; 5];
    % FA, T1w and T1w + FA
num_spect_component = 150;
    % number of eigen vectors in fingerprint
cell_pairwise_distance =
    cell(max_sib_type,length(array_modality_set));

for loop_i=1:length(array_modality_set)

    % compact fingerprint generation

```

---

```

    NxN_data_matrix =
    cell_NxN_data_matrix{array_modality_set(loop_i),1};
    mat_compact_fingerprint =
    compute_compact_fingerprint(NxN_data_matrix,num_spect_component);

    for loop_sib_type=1:max_sib_type
        sub_id_set = SAMPLE_cell_subID_and_sibpairID{loop_sib_type,1};
        pair_id_set =
        SAMPLE_cell_subID_and_sibpairID{loop_sib_type,2};
        [cell_pairwise_distance{loop_sib_type,loop_i}] =
        compute_pairwise_fingerprint_distance_siblings(sub_id_set,pair_id_set,mat_co
    end

    % get data for MZ, DZ, FS for a given modality
    cell_data = cell_pairwise_distance(:,loop_i);
    save_plot_name = [];
    modality = cell_modality(array_modality_set(loop_i));

    %count density histogram

    plot_compact_fingerprint_analysis_countdensity_histogram(cell_data,modality,save_

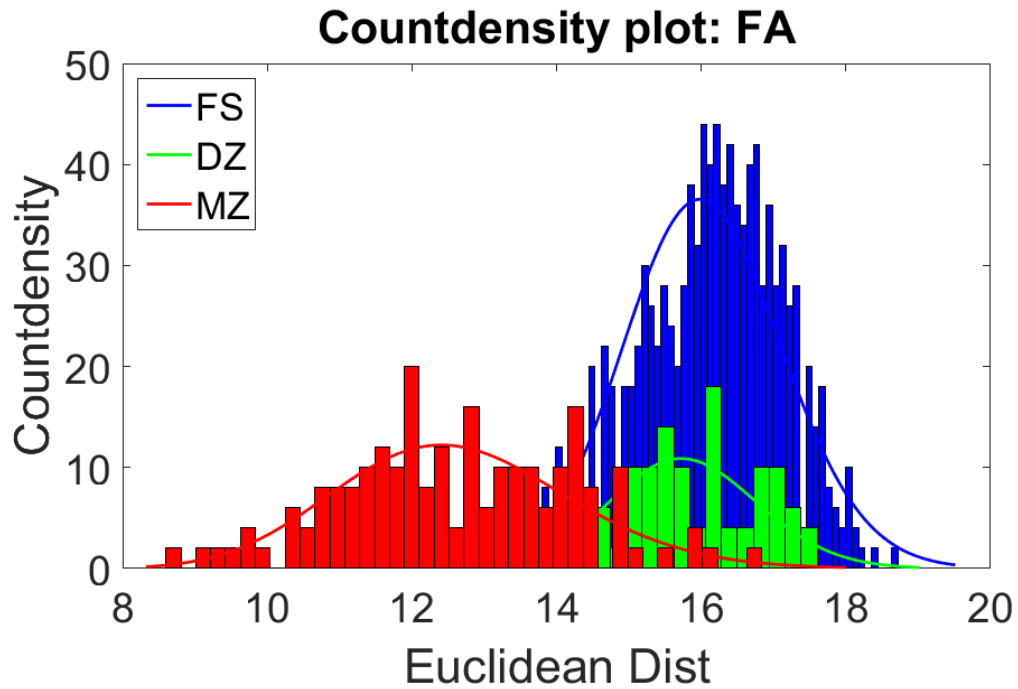
    %Plot probability normalized curves (gamma histogram fitting)

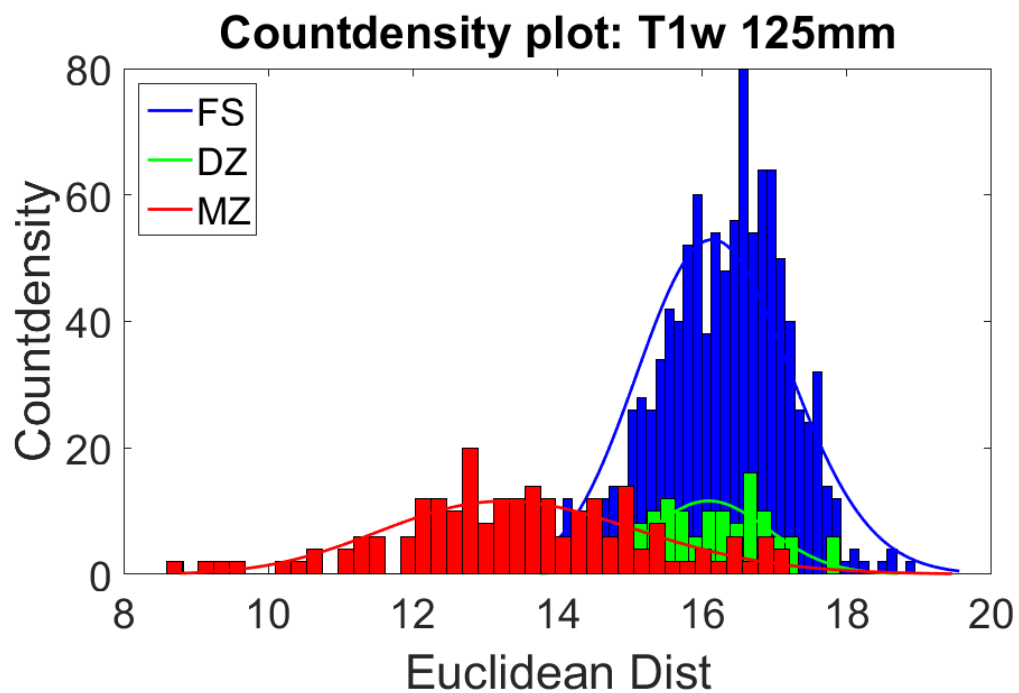
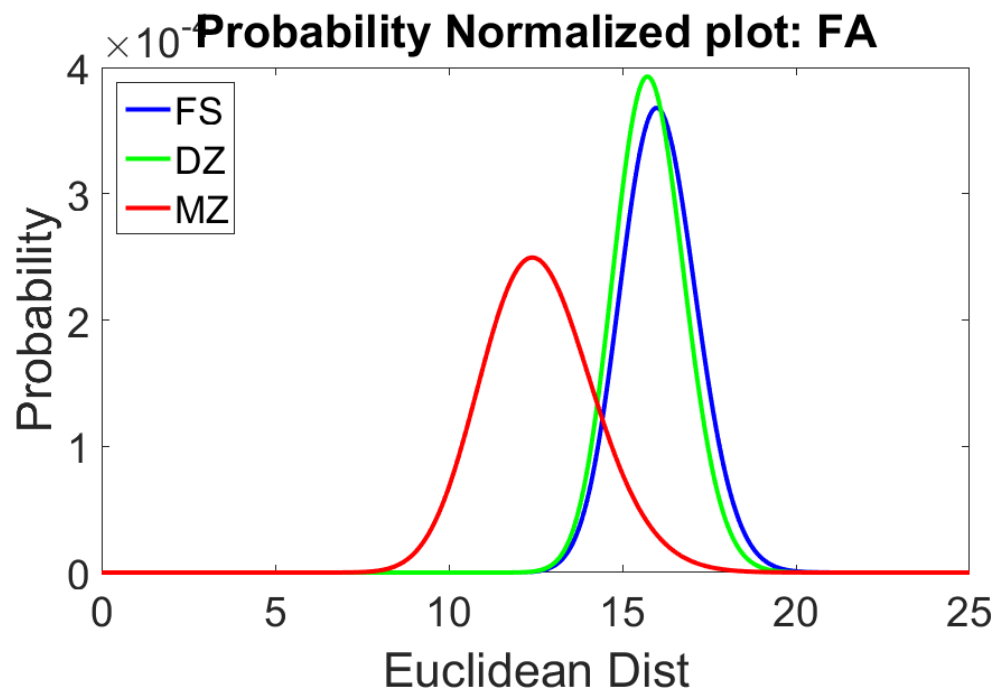
    plot_compact_fingerprint_analysis_probability_normalized_curves(cell_data,modality

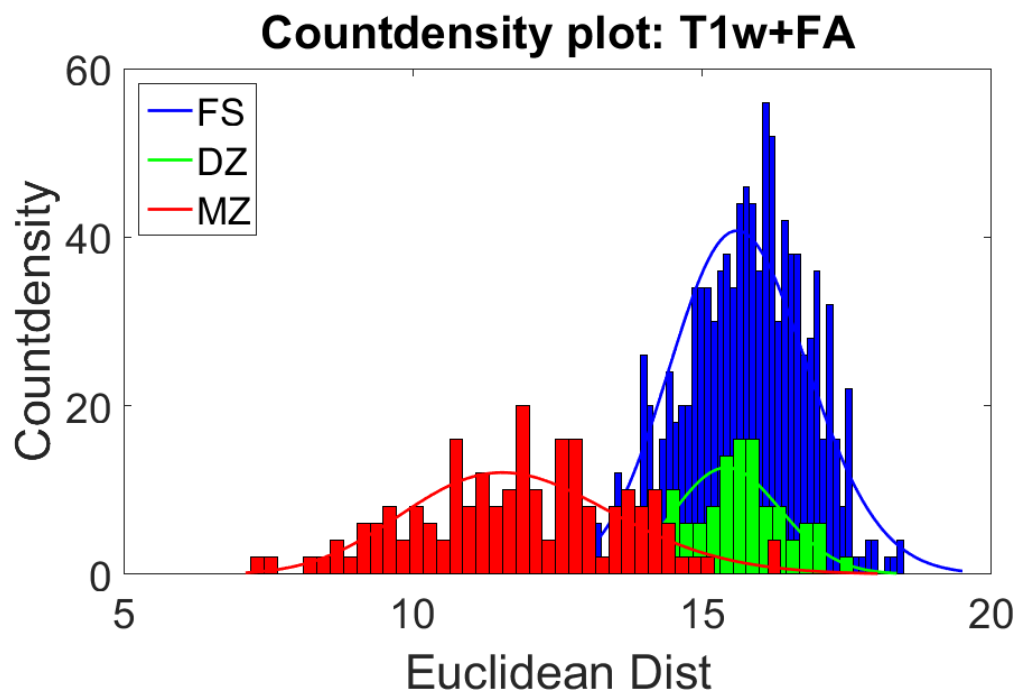
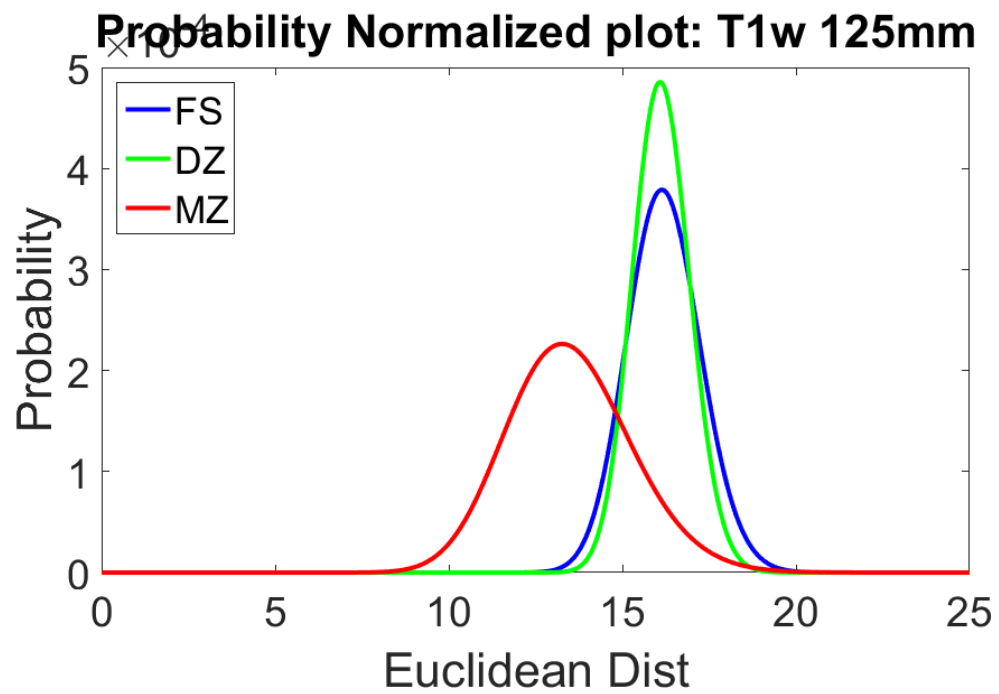
end

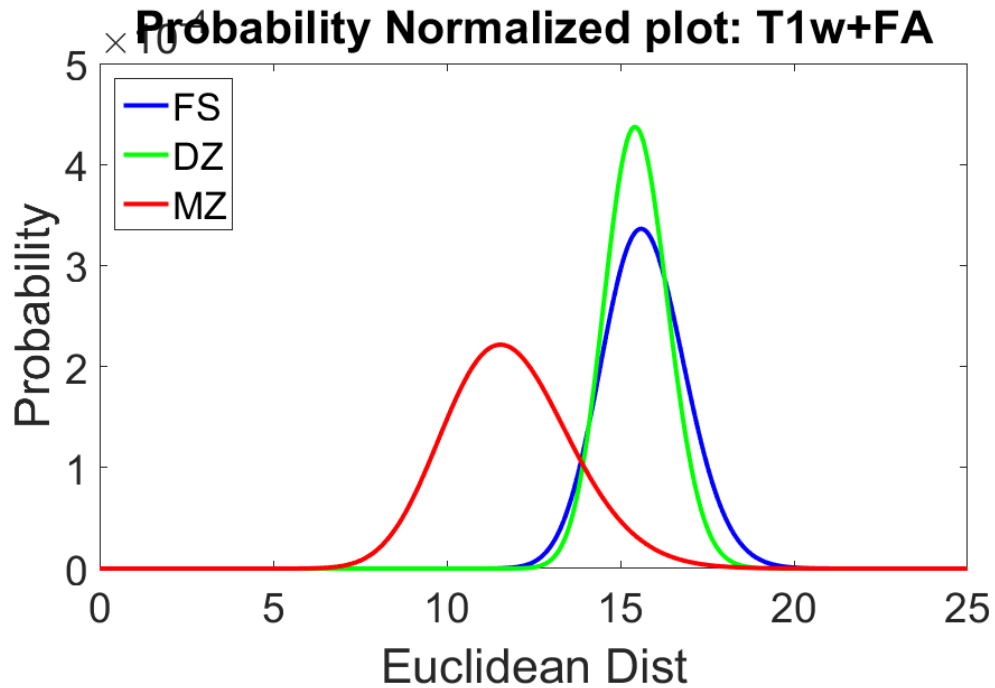
```

---









%%  
 %%%

## % Example 4: Rank Reterival Analysis: Mean Average Precision Table (Table 2, partial)

%%  
 %%%

```
%
% compute rank reterival measures for each sibling type and a given
% modality/combination
%       1. Mean Avg Precision and
%       2. mean recall@10
%
%       Additional outputs include detailed measures:
%       3. average precision array
%       4. recall@k    (k=1 to num_sub -1)
%       5. precision@k (k=1 to num_sub -1)
%
% Note: We can either use i) the subject proximity graphs directly or
% ii) compute pairwise Euclidean distance for compact fingerprints
%
```

---

```

flag_option_map = 1;
    % 1: subject proximity graphs 0: distance
array_modality_set = (1:9)' ;
num_spect_component = 150;
    % number of eigen vectors in fingerprint
cell_compact_fingerprint = cell(length(array_modality_set),1);

% rank reterival measures
mat_mean_avg_precision = zeros(length(array_modality_set),
    max_sib_type);
mat_mean_recall_at_10 = zeros(length(array_modality_set),
    max_sib_type);

% additional detailed measures
cell_array_avg_precision = cell(length(array_modality_set),
    max_sib_type);
cell_mat_recall_at_k = cell(length(array_modality_set), max_sib_type);
cell_mat_precision_at_k = cell(length(array_modality_set),
    max_sib_type);

for loop_i=1:length(array_modality_set)

    if(flag_option_map==1)
        % subject proximity graphs: Normalized Jaccard Similarity
        measures
        NxN_data_matrix =
        cell_NxN_data_matrix{array_modality_set(loop_i),1};
        flag_distance_or_similarity = 1;
        %: 0-distance; 1-similarity
    else
        % compact fingerprint generation
        NxN_data_matrix =
        compute_compact_fingerprint(cell_NxN_data_matrix{array_modality_set(loop_i),1},num_spect_component,
        flag_distance_or_similarity = 0;
        %: 0-distance; 1-similarity
    end

    knn_value = size(NxN_data_matrix,2)-1;
    % consider all subjects except self
    [nearest_neighbor_matrix] =
    compute_nearest_neighbor_matrix(NxN_data_matrix,knn_value,flag_distance_or_similarity);

    for loop_sib_type=1:max_sib_type
        sub_id_set =
        SAMPLE_cell_subID_and_sibpairID{loop_sib_type,1};
        pair_id_set =
        SAMPLE_cell_subID_and_sibpairID{loop_sib_type,2};

        [mat_mean_avg_precision(loop_i,loop_sib_type),mat_mean_recall_at_10(loop_i,loop_sib_type)] =
        compute_rank_reterival_measures(sub_id_set,pair_id_set,nearest_neighbor_matrix);
    end
end

```

---

---

## % Display mean average precision (MAP) values

```
data_matrix      = round(mat_mean_avg_precision,3);
cell_column      = cell_sibling_type(1:max_sib_type);
cell_row_label   = cell_modality ;

display_table_values(data_matrix, cell_column, cell_row_label);

sTable =
```

	<i>MZ</i>	<i>DZ</i>	<i>FS</i>
	_____	_____	_____
<i>FA</i>	0.964	0.219	0.16
<i>T1w 125mm</i>	0.831	0.136	0.121
<i>T2w 125mm</i>	0.879	0.173	0.132
<i>rfMRI</i>	0.968	0.352	0.205
<i>T1w+FA</i>	0.977	0.279	0.21
<i>T1w+rfMRI</i>	0.984	0.352	0.229
<i>FA+rfMRI</i>	0.99	0.439	0.276
<i>T1w+T2w+FA</i>	0.991	0.354	0.247
<i>T1w+T2w+FA+rfMRI</i>	0.997	0.546	0.371

%%%%%%%%%  
 %%%%%%%%%

## % Example 5: Mean Recall@10 values (Supplement material Fig Table 8, partial)

%%%%%%%%%  
 %%%%%%%%%

```
% Display mean recall @ 10 values
data_matrix      = round(mat_mean_recall_at_10,3);
cell_column      = cell_sibling_type(1:max_sib_type);
cell_row_label   = cell_modality ;

display_table_values(data_matrix, cell_column, cell_row_label);

sTable =
```

<i>MZ</i>	<i>DZ</i>	<i>FS</i>
_____	_____	_____

---

<i>FA</i>	0.992	0.421	0.307
<i>T1w 125mm</i>	0.903	0.317	0.22
<i>T2w 125mm</i>	0.954	0.349	0.249
<i>rfMRI</i>	0.975	0.524	0.369
<i>T1w+FA</i>	0.996	0.508	0.372
<i>T1w+rfMRI</i>	0.996	0.635	0.438
<i>FA+rfMRI</i>	0.996	0.69	0.47
<i>T1w+T2w+FA</i>	1	0.579	0.429
<i>T1w+T2w+FA+rfMRI</i>	1	0.817	0.603

%%%%%%%%%

%%%%%%%%%

## % Example 6: Mean Recall@k Plots (Supplement material Fig 1.)

%%%%%%%%%

%%%%%%%%%

```
temp_knn_value =50;
    % number of nearest neighbors or k value

% get mean recall@k for k=1,2,...,temp_knn_value
array_modality_set = [ 1; 2; 4; 5; 9;];
    % see cell_modality for index reference
mat_mean_recall_at_k =
    zeros(length(array_modality_set),temp_knn_value,max_sib_type);

% compute mean recall@k values using recal@k matrix
for loop_i=1:length(array_modality_set)
    for loop_sib_type =1:max_sib_type
        temp_mat =
            cell_mat_recall_at_k{array_modality_set(loop_i),loop_sib_type}
        (:,1:50);
        mat_mean_recall_at_k(loop_i,:,loop_sib_type) = mean(temp_mat);
    end
end

%plot for each twin/sibling type
for loop_sib_type =1:max_sib_type

    data_mat  = mat_mean_recall_at_k(:, :, loop_sib_type);
    x_array   = (1:50)';

    figure_title = ['mean recall@k plot: ',
        cell_sibling_type{loop_sib_type}];
```



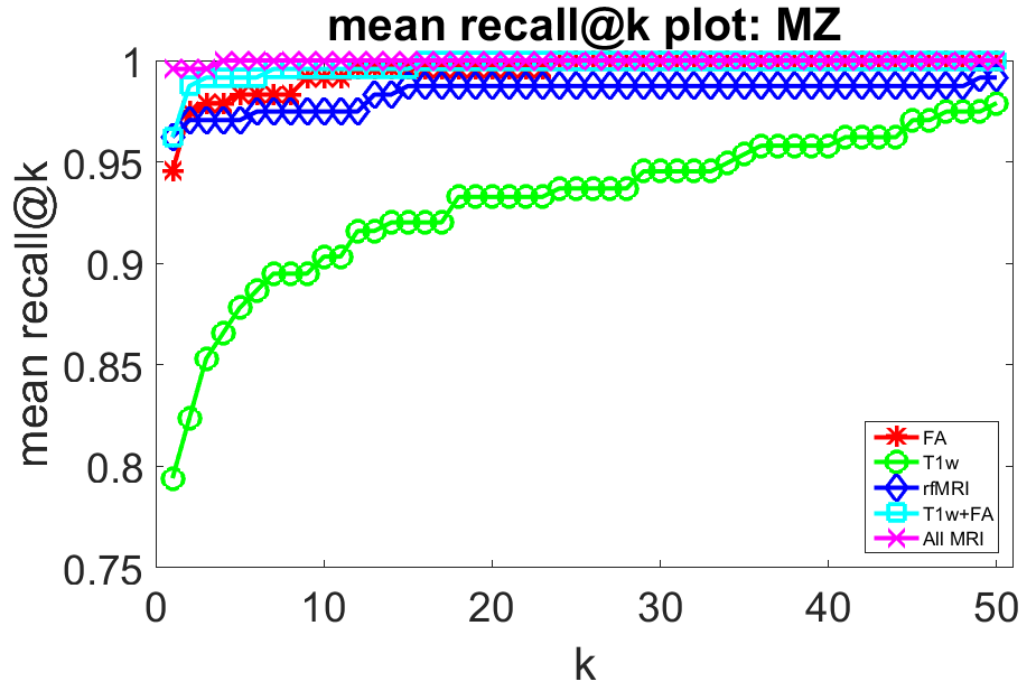
---

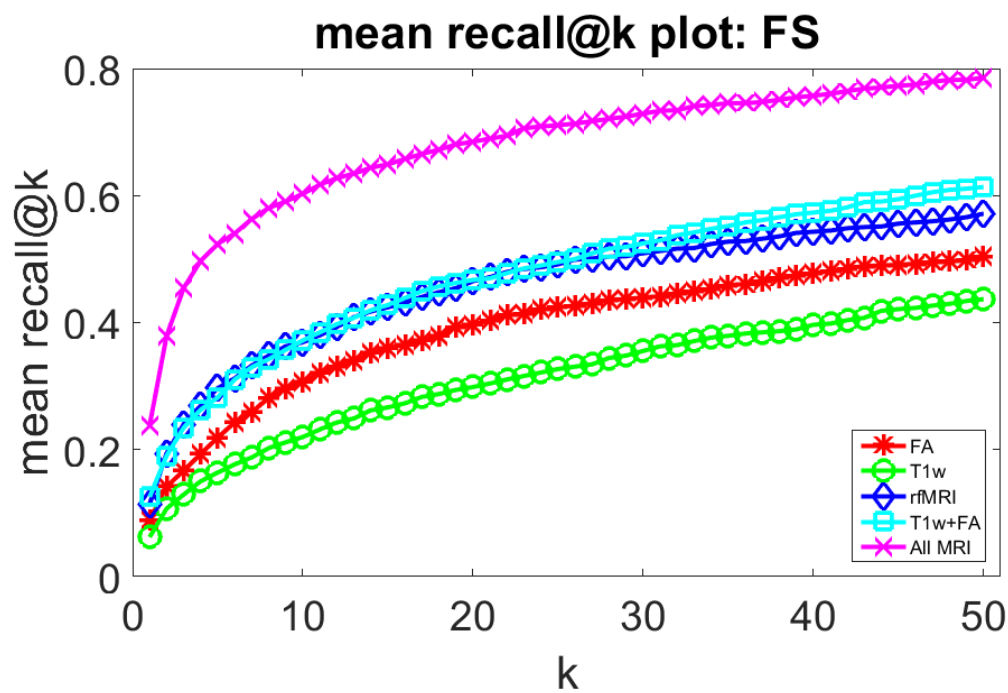
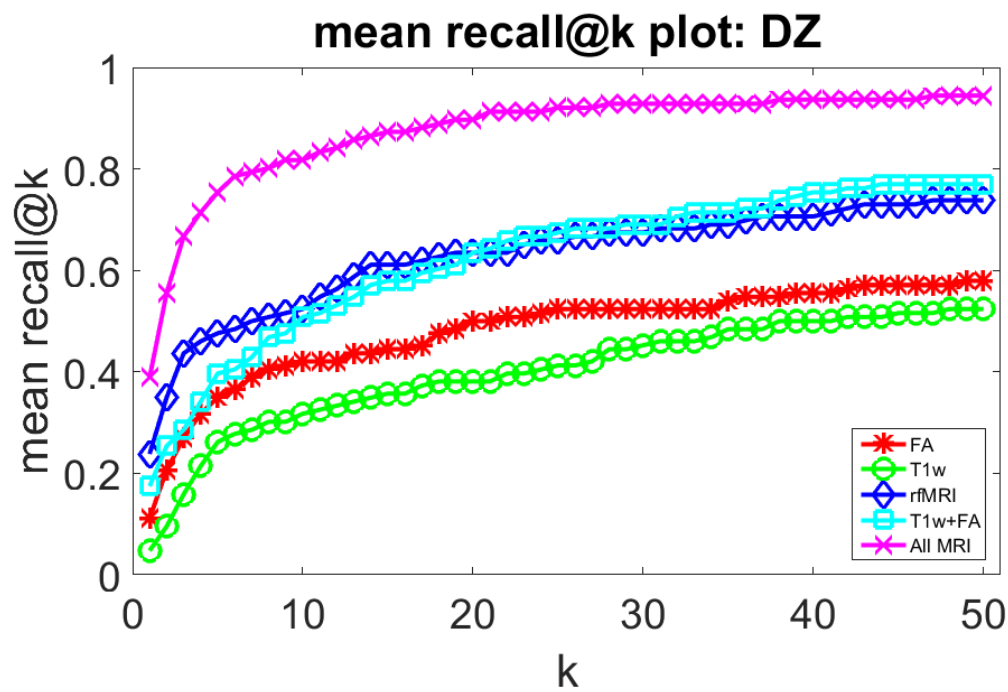
```

cell_label{1} = 'k';
cell_label{2} = 'mean recall@k';
leg_location = 'southeast';
cell_legend = {'FA'; 'T1w'; 'rfMRI'; 'T1w+FA'; 'All MRI'};
save_plot_name = [];
    % assign filename to save it else empty

plot_mean_recall_at_k(data_mat,x_array,figure_title,cell_legend,leg_location,cell_label{1},cell_label{2})
end

```





---

%%%%%%%%%%  
%%%%%%%%

## % Example 7: Relative informativeness Table: MZ, DZ, FS (Table 3)

%%%%%%%%%%  
%%%%%%%%

```
% Compute relative informativeness

relative_identification_knn_value = 10;
    % number of nearest neighbors for realtive identification

    % identification considered success within these many neighbors
    only

cell_compare_modality1_vs_modality2 = { 'T1w vs T2w';
                                         'T1w vs FA';
                                         'T1w vs rfMRI';
                                         'FA vs rfMRI';
                                         'T1w vs T1w+T2w+FA+rfMRI';
                                         'T2w vs T1w+T2w+FA+rfMRI';
                                         'FA vs T1w+T2w+FA+rfMRI';
                                         'rfMRI vs T1w+T2w+FA+rfMRI';
                                         };

% hard code the modalities to be compared
% indices are based on cell_modality
array_modality1_index = [ 2; 2; 2; 1; 2; 3; 1; 4; ];
array_modality2_index = [ 3; 1; 4; 4; 9; 9; 9; 9; ];
mat_relative_identification_percent =
    zeros(length(array_modality1_index),4,max_sib_type);

for loop_i=1:length(array_modality1_index)

    if(flag_option_map==1)
        % subject proximity graphs: Normalized Jaccard Similarity
        measures
        NxN_data_matrix_1 =
        cell_NxN_data_matrix{array_modality1_index(loop_i),1};
        NxN_data_matrix_2 =
        cell_NxN_data_matrix{array_modality2_index(loop_i),1};
        flag_distance_or_similarity = 1;
        %: 0-distance; 1-similarity
    else
        % compact fingerprint generation
```

---

```

        NxN_data_matrix_1 =
compute_compact_fingerprint(cell_NxN_data_matrix{array_modality1_index(loop_i),1}
        NxN_data_matrix_2 =
compute_compact_fingerprint(cell_NxN_data_matrix{array_modality2_index(loop_i),1}
        flag_distance_or_similarity = 0;
        %: 0-distance; 1-similarity
    end

    nearest_neighbor_matrix_mod1=
compute_nearest_neighbor_matrix(NxN_data_matrix_1,relative_identification_knn_val
    nearest_neighbor_matrix_mod2=
compute_nearest_neighbor_matrix(NxN_data_matrix_2,relative_identification_knn_val

    for loop_sib=1:max_sib_type
        sub_id_set = SAMPLE_cell_subID_and_sibpairID{loop_sib,1};
        pair_id_set = SAMPLE_cell_subID_and_sibpairID{loop_sib,2};
        [mat_relative_identification_percent(loop_i,1:4,loop_sib)]
    =
    compute_relative_identification_percentage(sub_id_set,pair_id_set,nearest_neighbo
    end
end

```

## % Display realtive identification percentages: MZ, DZ, FS tables

```

cell_relative_comparison_order = { 'Both'; 'Mod1'; 'Mod2';
'None'; };
cell_column = cell_relative_comparison_order;
cell_row_label = cell_compare_modality1_vs_modality2 ;

for loop_sib=1:max_sib_type
    % display for each twin/sibling type
    disp(['Relative Identification percentage for : '
cell_sibling_type{loop_sib}]);
    data_matrix
    =round(mat_relative_identification_percent(:, :, loop_sib),2);
    display_table_values(data_matrix, cell_column, cell_row_label);
end

```

*Relative Identification percentage for : MZ*

*sTable =*

	<i>Both</i>	<i>Mod1</i>	<i>Mod2</i>	<i>None</i>
<i>T1w vs T2w</i>	<i>85.71</i>	<i>4.62</i>	<i>9.66</i>	<i>0</i>
<i>T1w vs FA</i>	<i>90.34</i>	<i>0</i>	<i>8.82</i>	<i>0.84</i>
<i>T1w vs rfMRI</i>	<i>88.24</i>	<i>2.1</i>	<i>9.24</i>	<i>0.42</i>
<i>FA vs rfMRI</i>	<i>96.64</i>	<i>2.52</i>	<i>0.84</i>	<i>0</i>
<i>T1w vs T1w+T2w+FA+rfMRI</i>	<i>90.34</i>	<i>0</i>	<i>9.66</i>	<i>0</i>
<i>T2w vs T1w+T2w+FA+rfMRI</i>	<i>95.38</i>	<i>0</i>	<i>4.62</i>	<i>0</i>

---

<i>FA vs T1w+T2w+FA+rfMRI</i>	99.16	0	0.84	0
<i>rfMRI vs T1w+T2w+FA+rfMRI</i>	97.48	0	2.52	0

Relative Identification percentage for : DZ

*sTable* =

	<i>Both</i>	<i>Mod1</i>	<i>Mod2</i>	<i>None</i>
	<hr/>	<hr/>	<hr/>	<hr/>
<i>T1w vs T2w</i>	15.87	15.87	19.05	49.21
<i>T1w vs FA</i>	15.87	15.87	26.19	42.06
<i>T1w vs rfMRI</i>	18.25	13.49	34.13	34.13
<i>FA vs rfMRI</i>	26.19	15.87	26.19	31.75
<i>T1w vs T1w+T2w+FA+rfMRI</i>	28.57	3.17	53.17	15.08
<i>T2w vs T1w+T2w+FA+rfMRI</i>	33.33	1.59	48.41	16.67
<i>FA vs T1w+T2w+FA+rfMRI</i>	39.68	2.38	42.06	15.87
<i>rfMRI vs T1w+T2w+FA+rfMRI</i>	49.21	3.17	32.54	15.08

Relative Identification percentage for : FS

*sTable* =

	<i>Both</i>	<i>Mod1</i>	<i>Mod2</i>	<i>None</i>
	<hr/>	<hr/>	<hr/>	<hr/>
<i>T1w vs T2w</i>	10.26	11.72	14.65	63.37
<i>T1w vs FA</i>	10.62	11.36	20.05	57.97
<i>T1w vs rfMRI</i>	9.52	12.45	27.38	50.64
<i>FA vs rfMRI</i>	12.91	17.77	23.99	45.33
<i>T1w vs T1w+T2w+FA+rfMRI</i>	18.96	3.02	41.3	36.72
<i>T2w vs T1w+T2w+FA+rfMRI</i>	21.98	2.93	38.28	36.81
<i>FA vs T1w+T2w+FA+rfMRI</i>	25.64	5.04	34.62	34.71
<i>rfMRI vs T1w+T2w+FA+rfMRI</i>	34.89	2.01	25.37	37.73

%%%

%%%

%%%

%%%

Published with MATLAB® R2015b