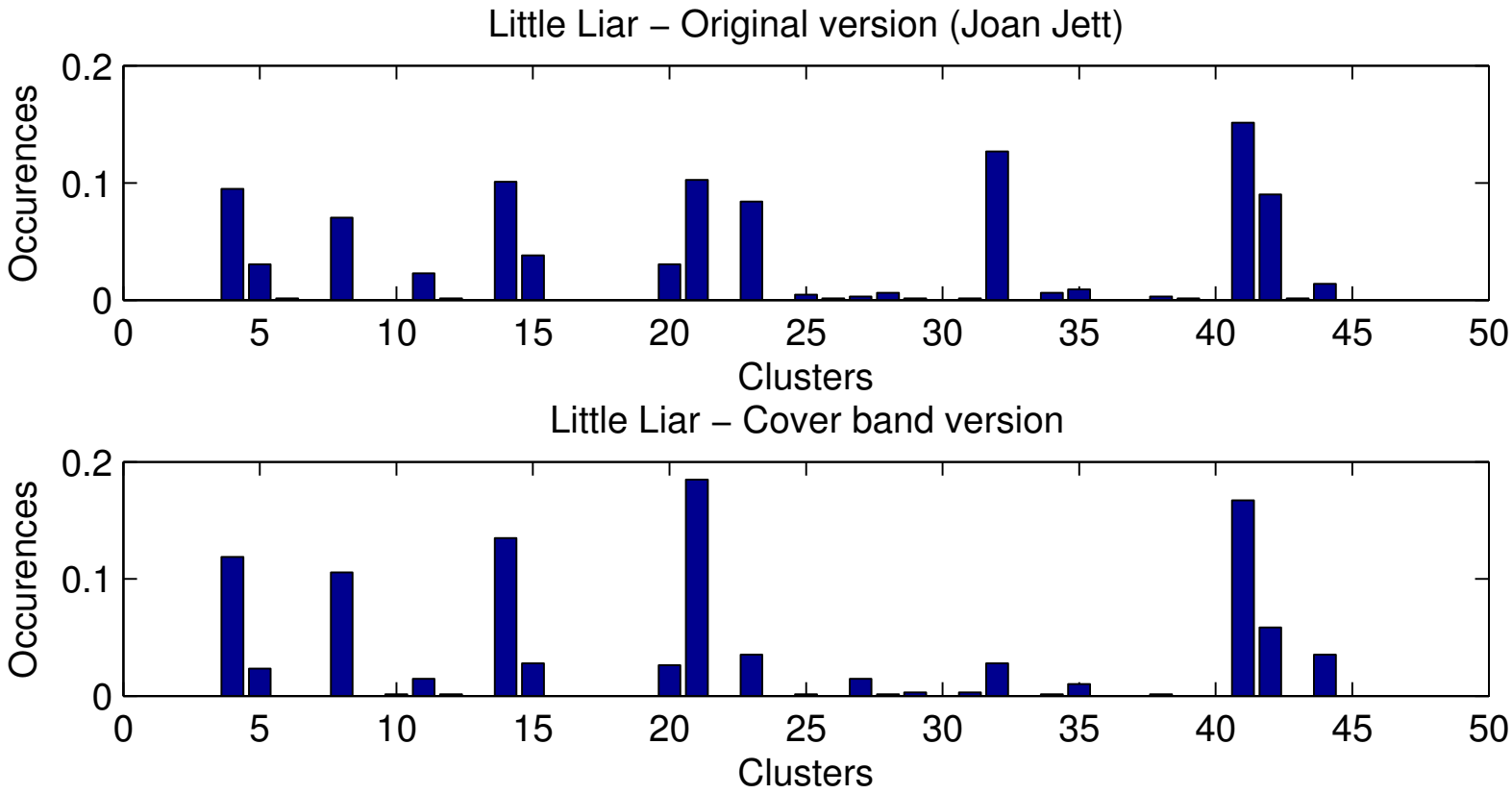
A Combining Approach to Cover Song Identification

Julien OSMALSKYJ

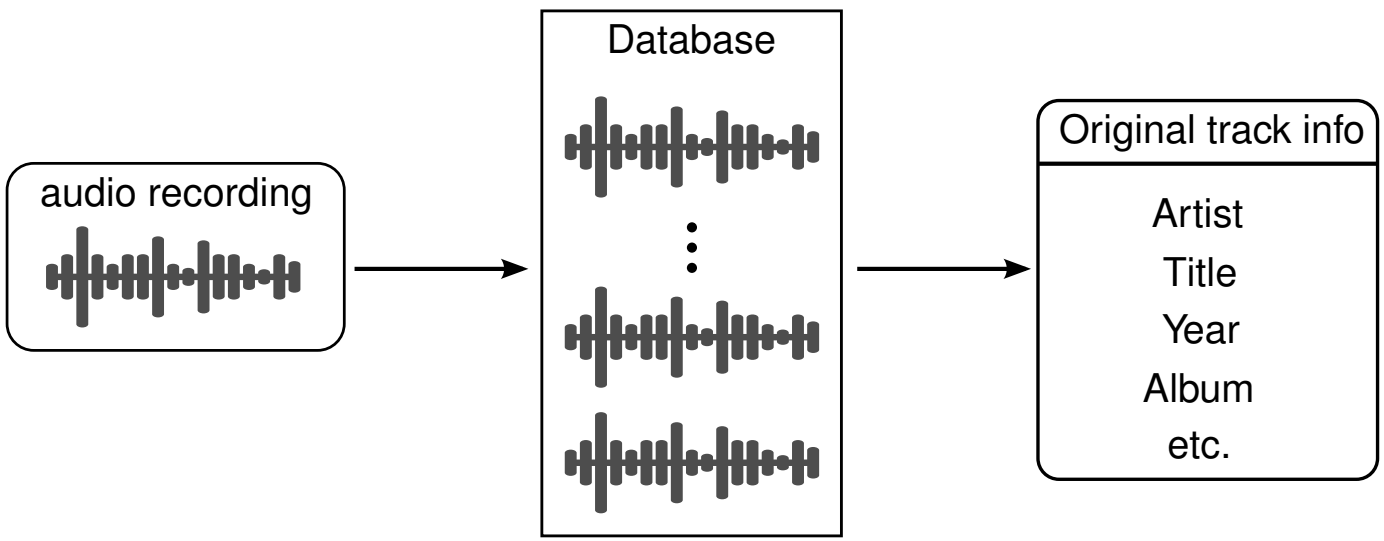
University of Liège
Montefiore Institute
Department of Electrical Engineering and Computer Science
Belgium

October 5, 2017

Can we quantify how a musical performance of an existing song is close to the original song ?



Little Liar – Original version (Joan Jett)
Occurences
0.2
0.1
0
0
5
10
15
20
25
30
35
40
45
50
Clusters
Little Liar – Cover band version
Occurences
0.2
0.1
0
0
5
10
15
20
25
30
35
40
45
50
Clusters

How could we identify an unknown cover in a large database of songs ?



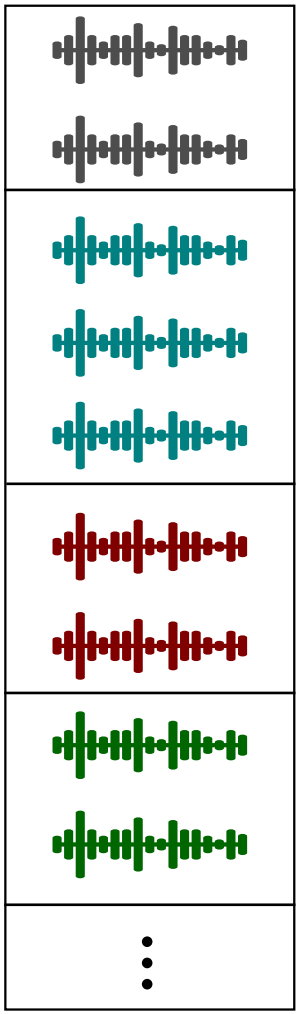
How is a cover song defined ?

Cover Song

A cover is a performance of a work that is not an original, performed by an artist different from the artist performing the original performance.

How about a musical definition of a cover song ?

How can we do it ?

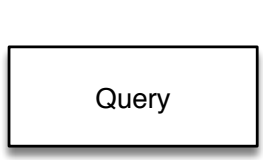


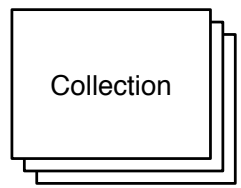
- Create a database of songs grouped by cover versions based on our human perception of what cover songs are.
- Design algorithms that match that definition of cover songs.
- This is the field of Cover Song Identification (CSI).

How do we do it in practice ?

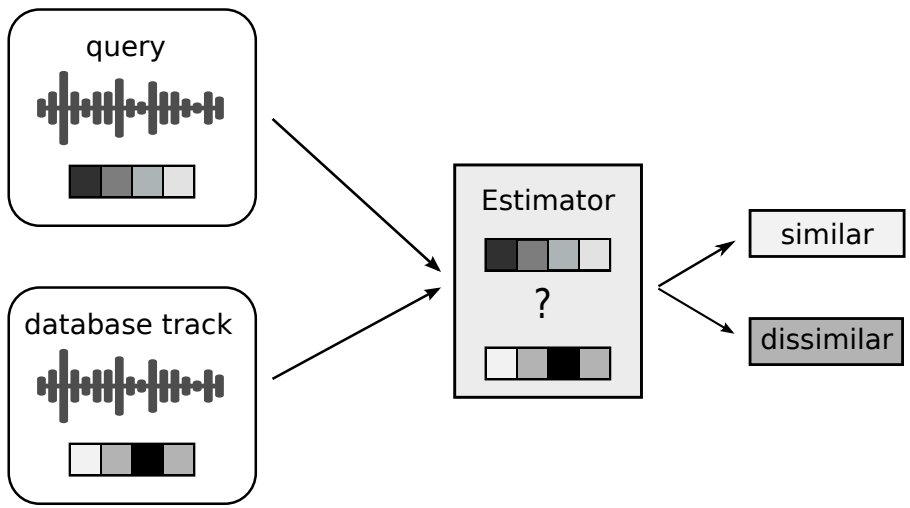
Compare a query to a collection of tracks

We need an audio query and a reference collection of audio musical tracks.



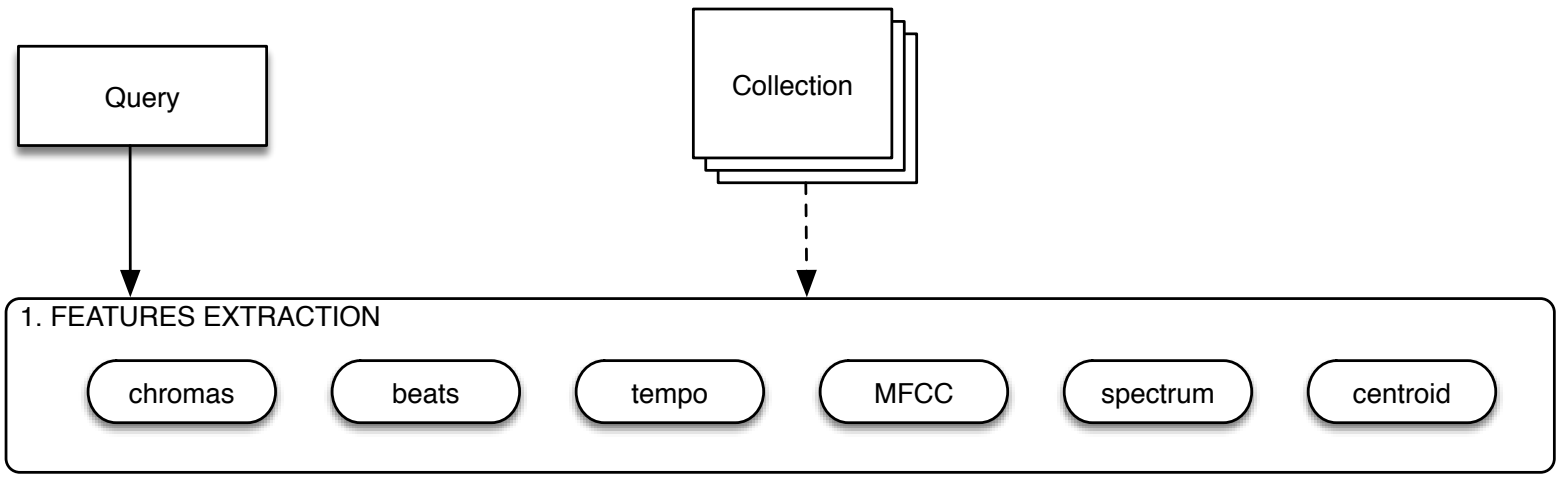


The query and all the tracks of the collection must be described with the same representation.



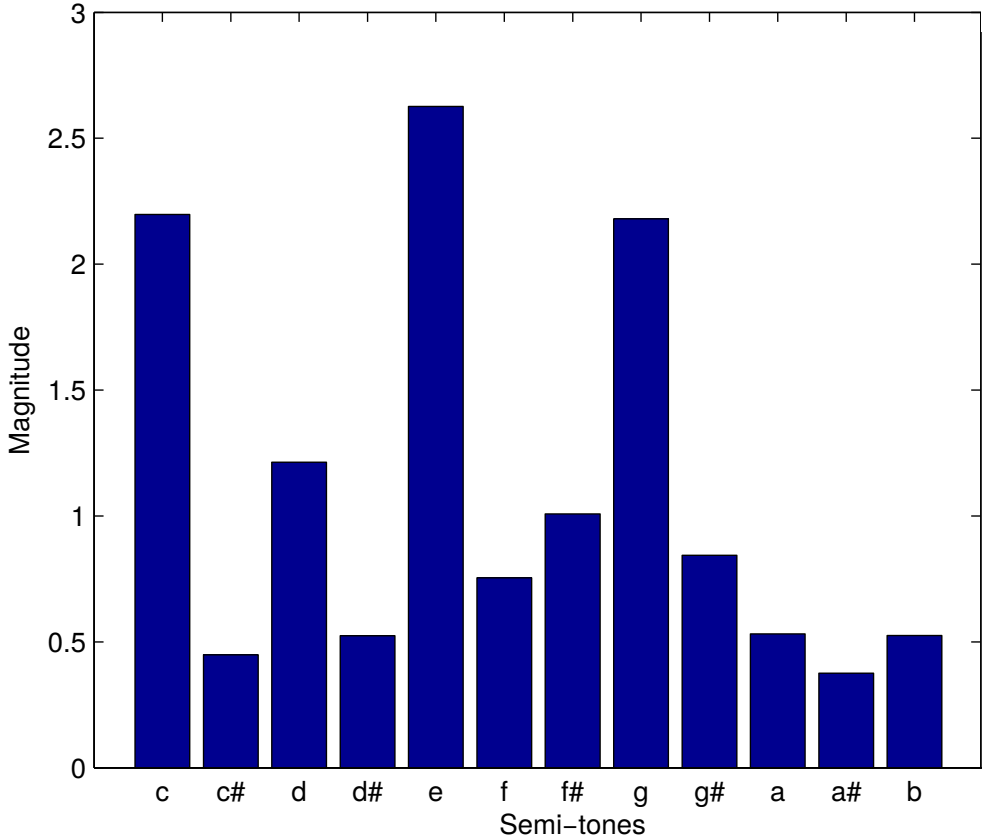
Extract audio features

Representation is given by audio features that are extracted from the signal.



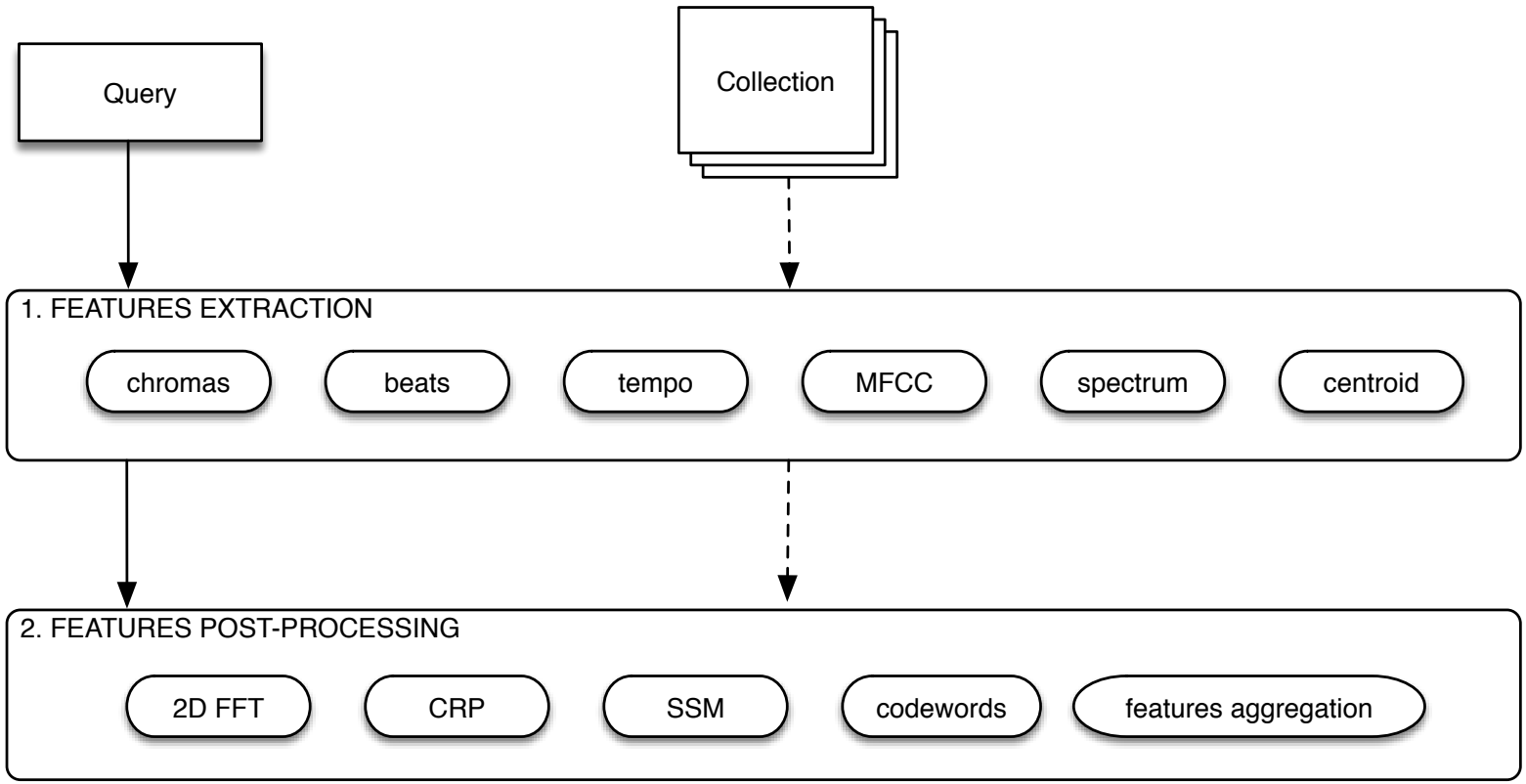
Audio features are computed from the raw signal and characterize the musical content of the music.

Chroma features



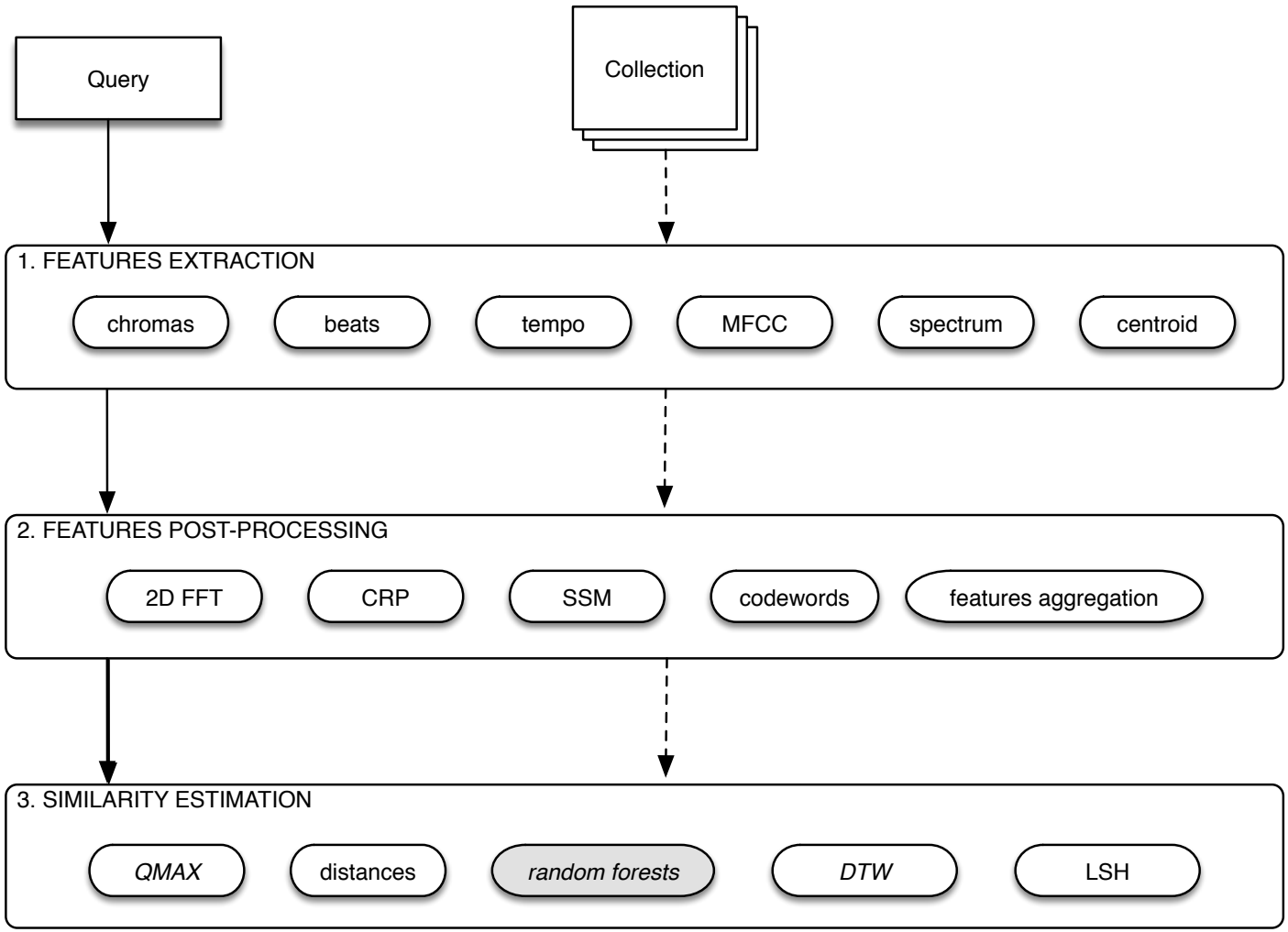
Map octave-folded frequency bands to 12 pitch classes in the chromatic scale.

Process features



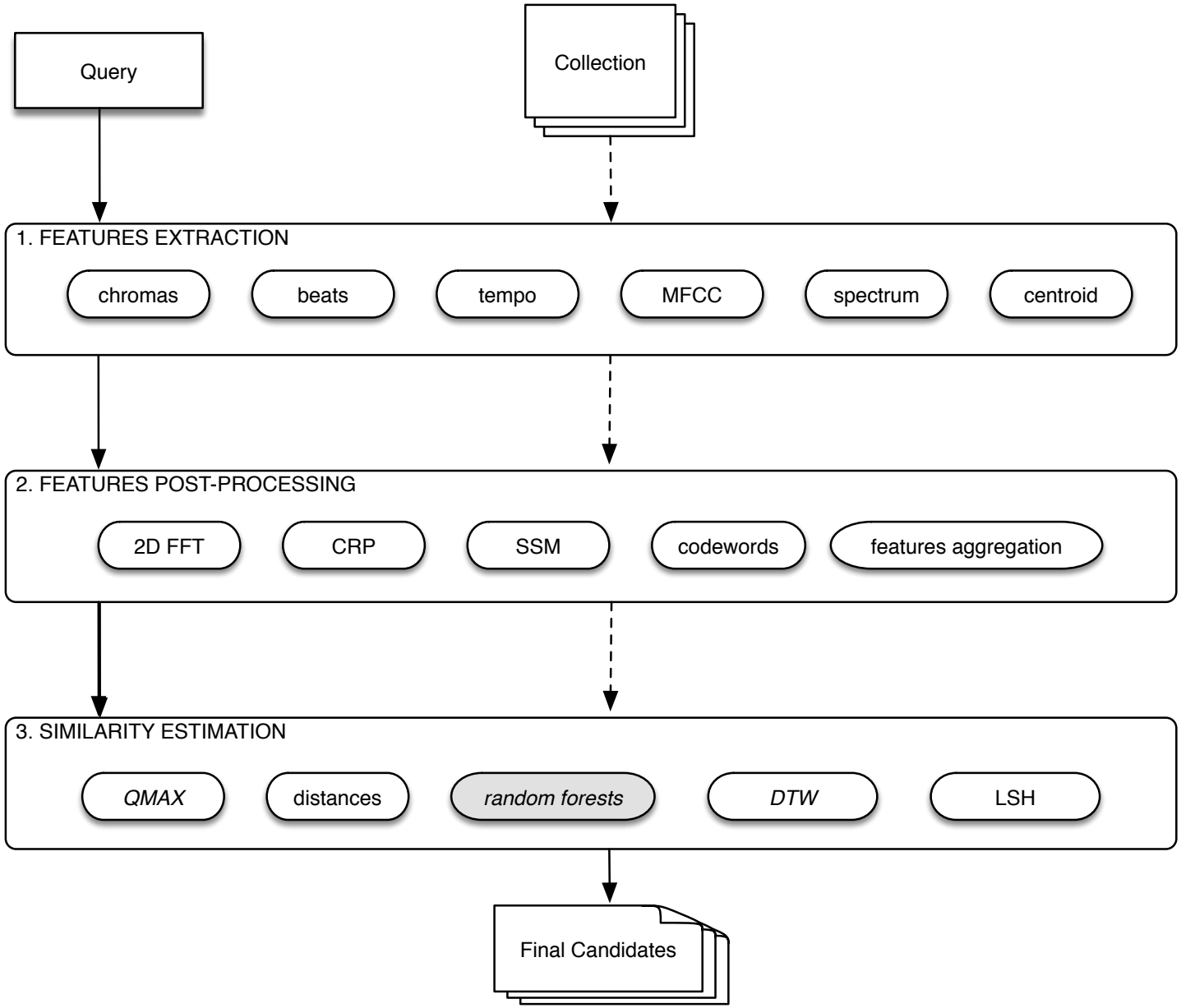
Features are processed to transform the data so that it is easier to compare the tracks.

Compare sets of features

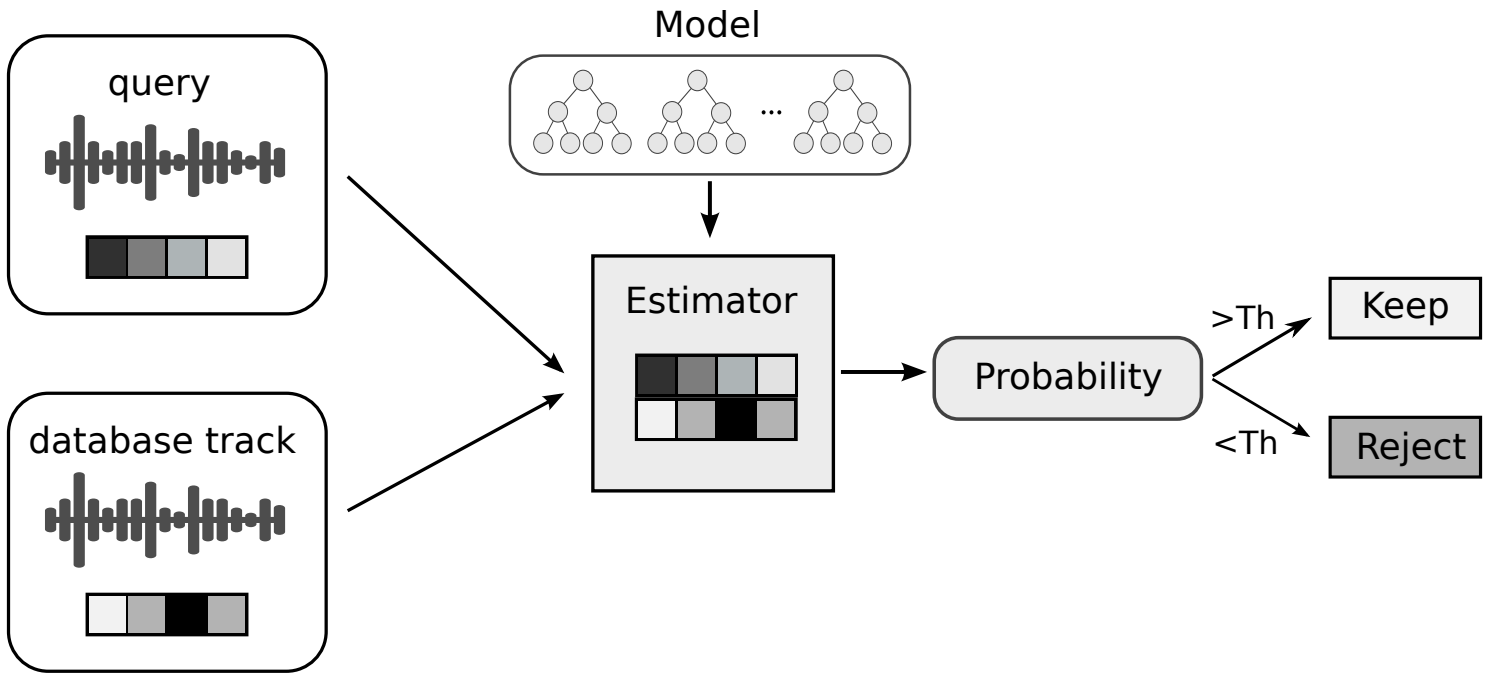


Compare the query to each track of the collection and compute a similarity score.

Return set of candidates

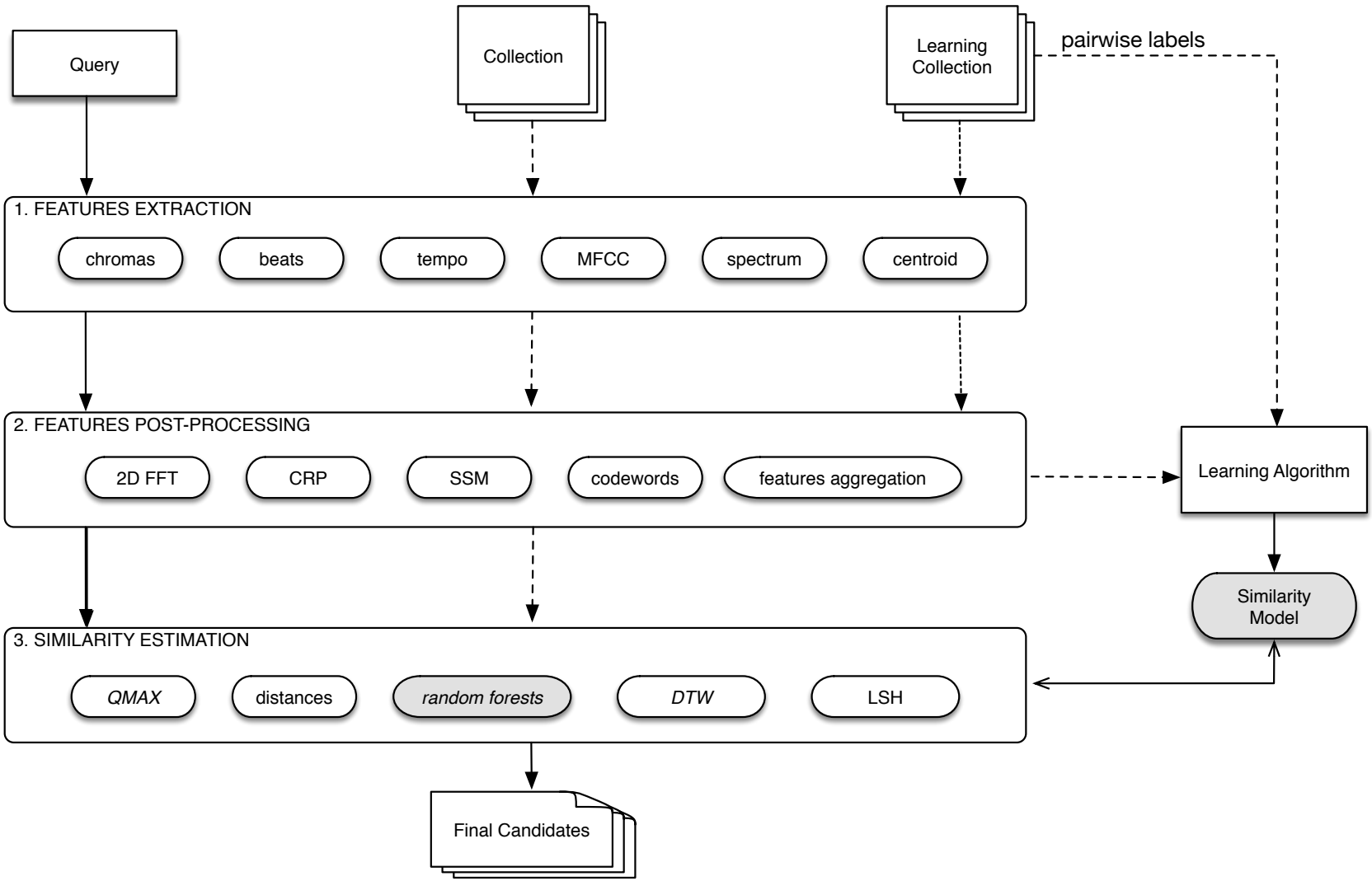


Possible use of machine learning



Similarity estimation can be done through a learned distance. A learning collection is required to learn the model.

Pipeline with machine learning



How do current systems perform ?

Several existing systems designed and evaluated in the litterature.

Difficult to compare the performance:

- Different databases used (often small in size)
- Use of different implementations of the features
- Different evaluation procedures
- Results reported using different metrics

No existing comparison of the performance of CSI systems.

Contributions

1. Proposition of a new evaluation space for comparing the performance of CSI systems

2. Evaluation of 10 systems on a large database using the proposed space

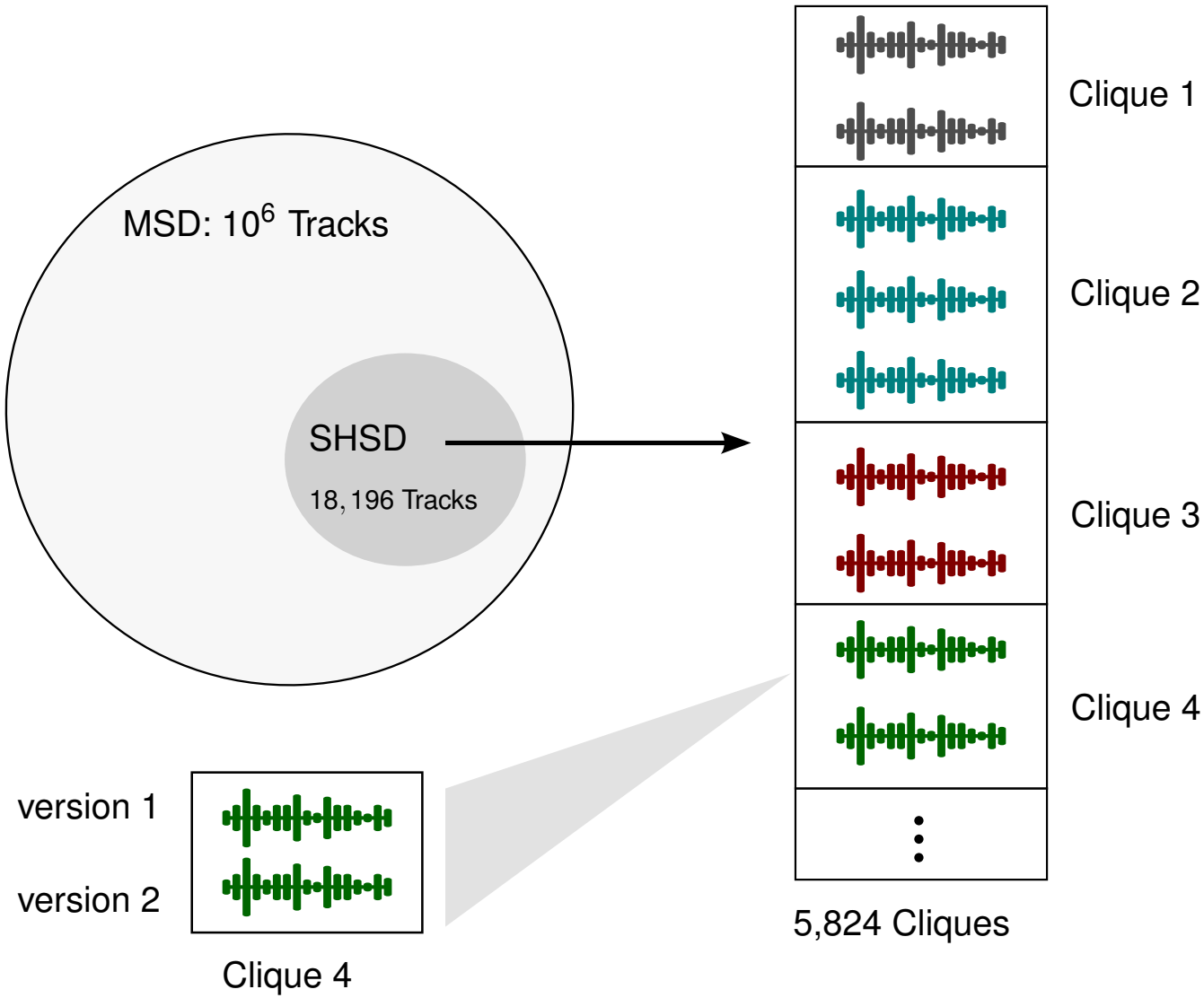
Evaluating CSI systems

To evaluate a cover song identification system we need

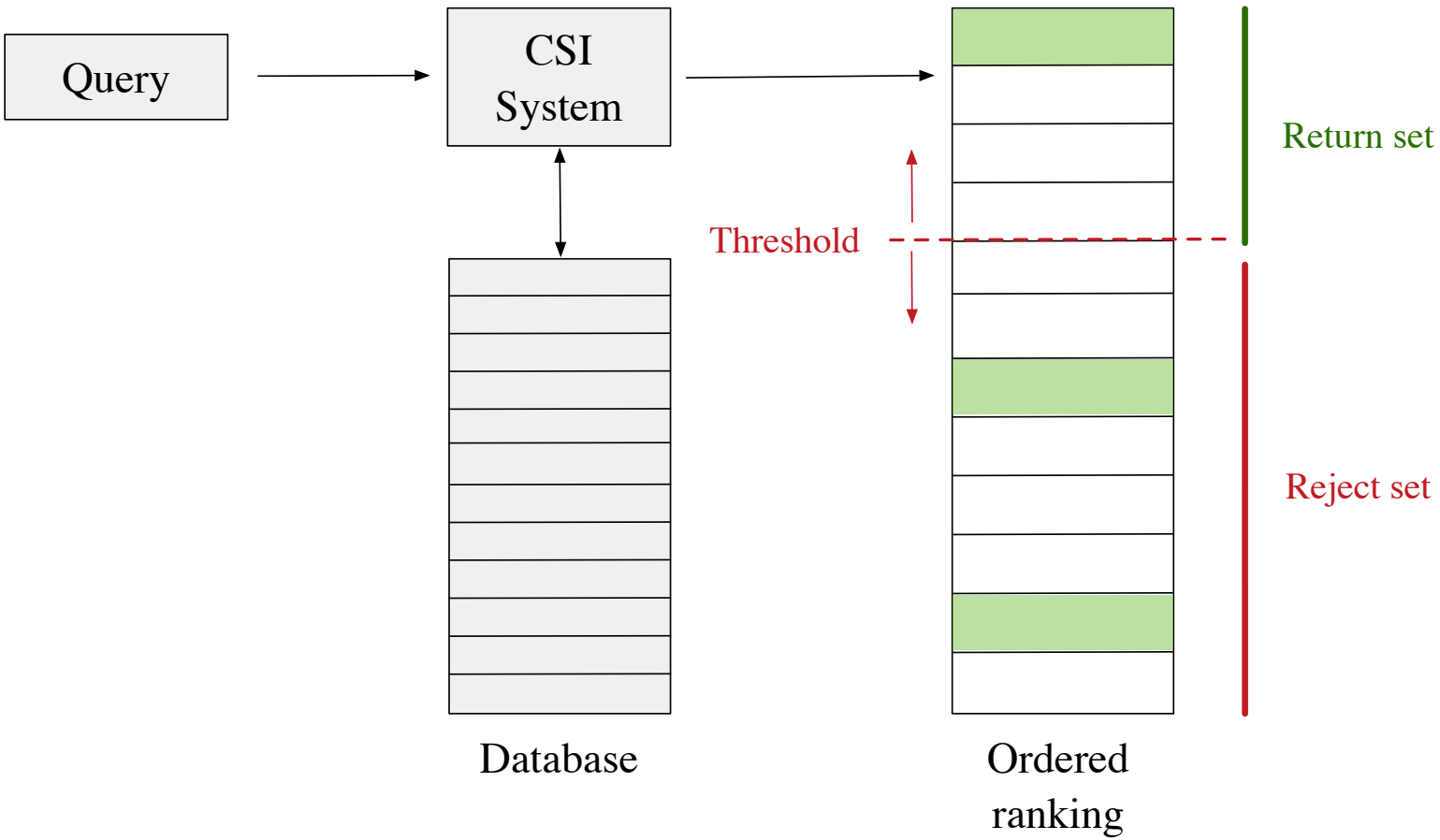
- An evaluation database
- A ground-truth
- An evaluation procedure

The database should be as large as possible to reflect the performance on a large-scale.

The Second Hand Song Dataset

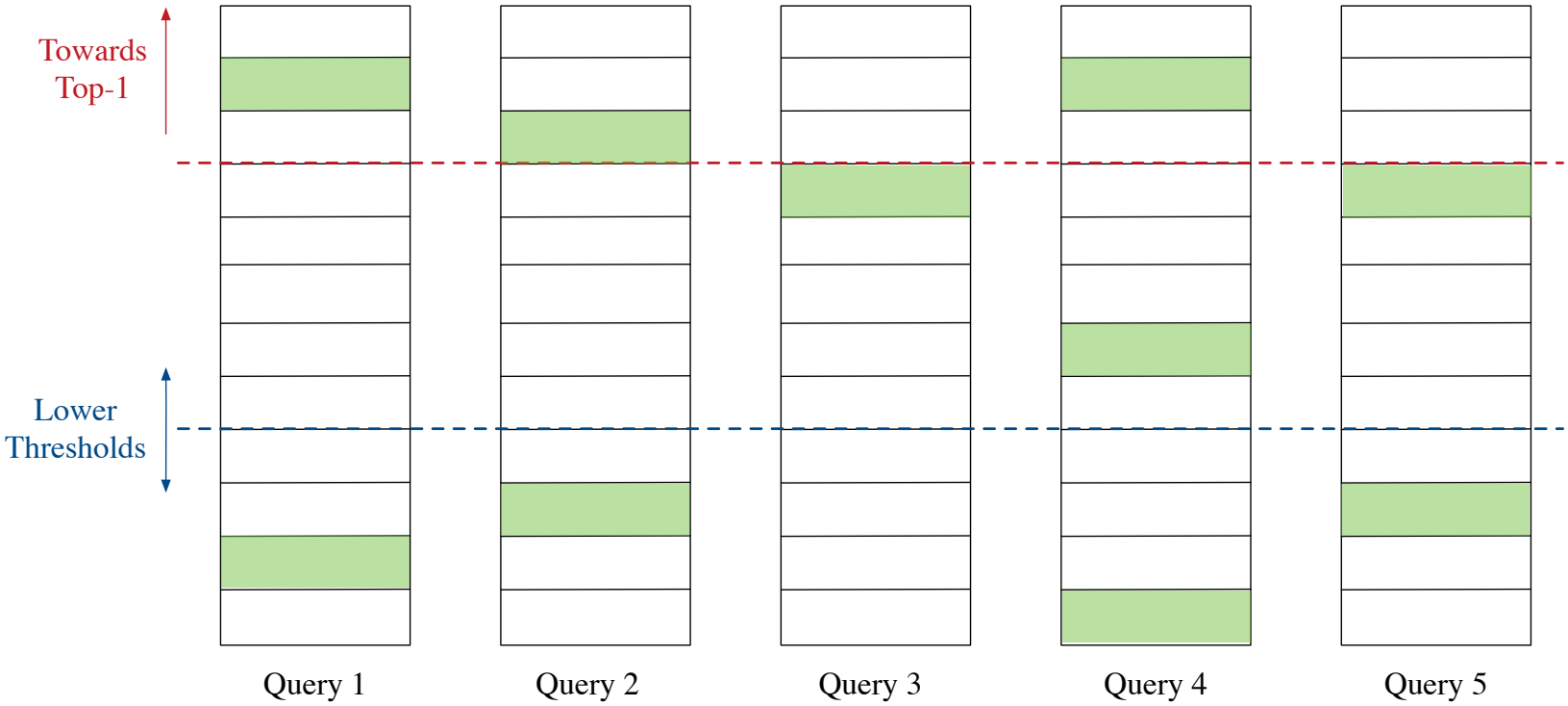


Evaluation procedure



Take each track as a query and compare it to the remaining tracks of the database.

Performance scores



CSI scores such as MRR, MAP, TOP-1 consider the performance near the top.

Scores interpretation

Performance scores are associated to a specific use case.

Use case for MRR, MAP, TOP-1

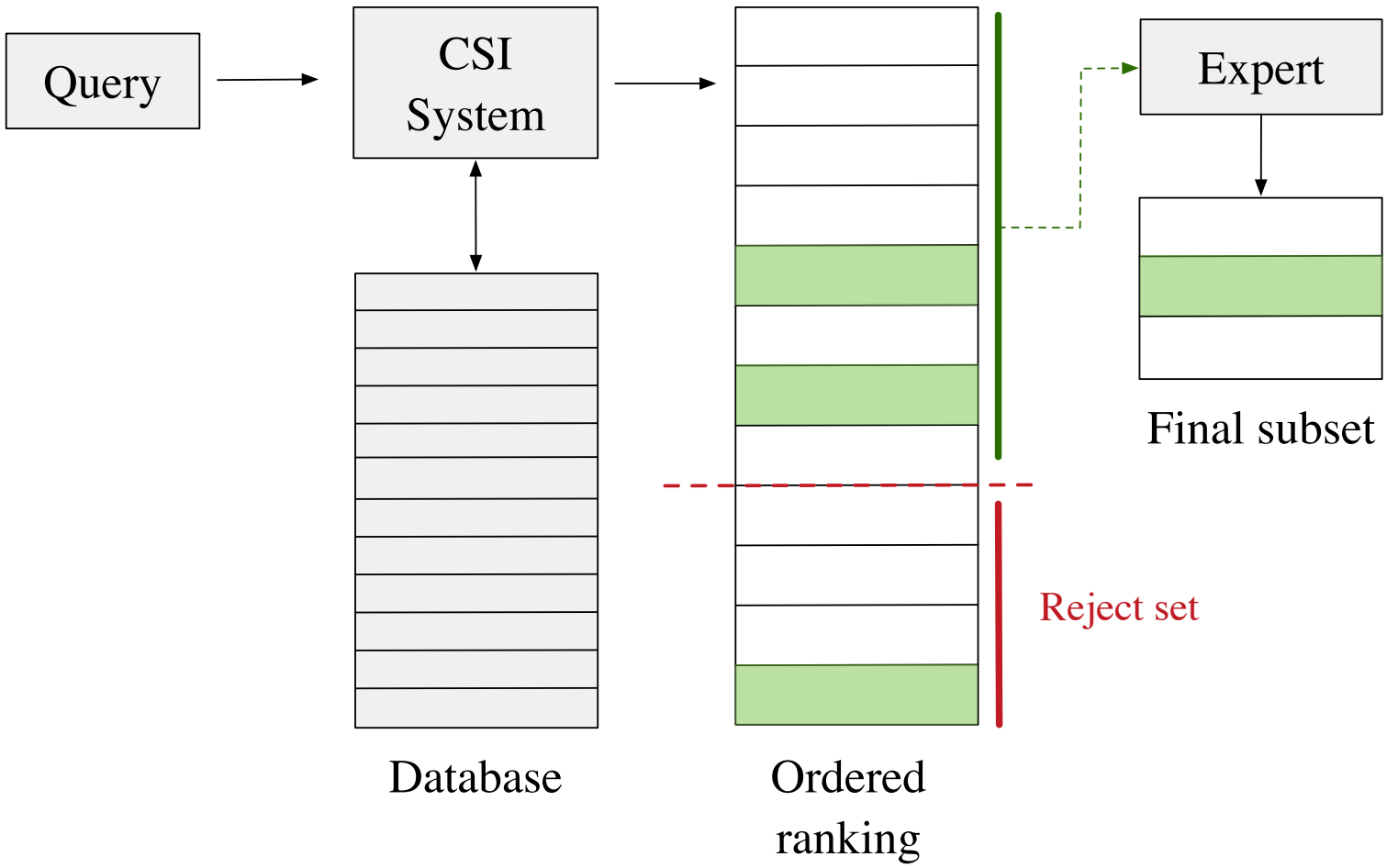
Identify all versions for all queries close to the top of the ranking

- MRR increases when more queries have 1 version identified near the top.
- TOP-1 increases when more queries have 1 version identified at the first position.
- MAP increases when more versions are identified near the top.

What is the best performance achievable by a system ?

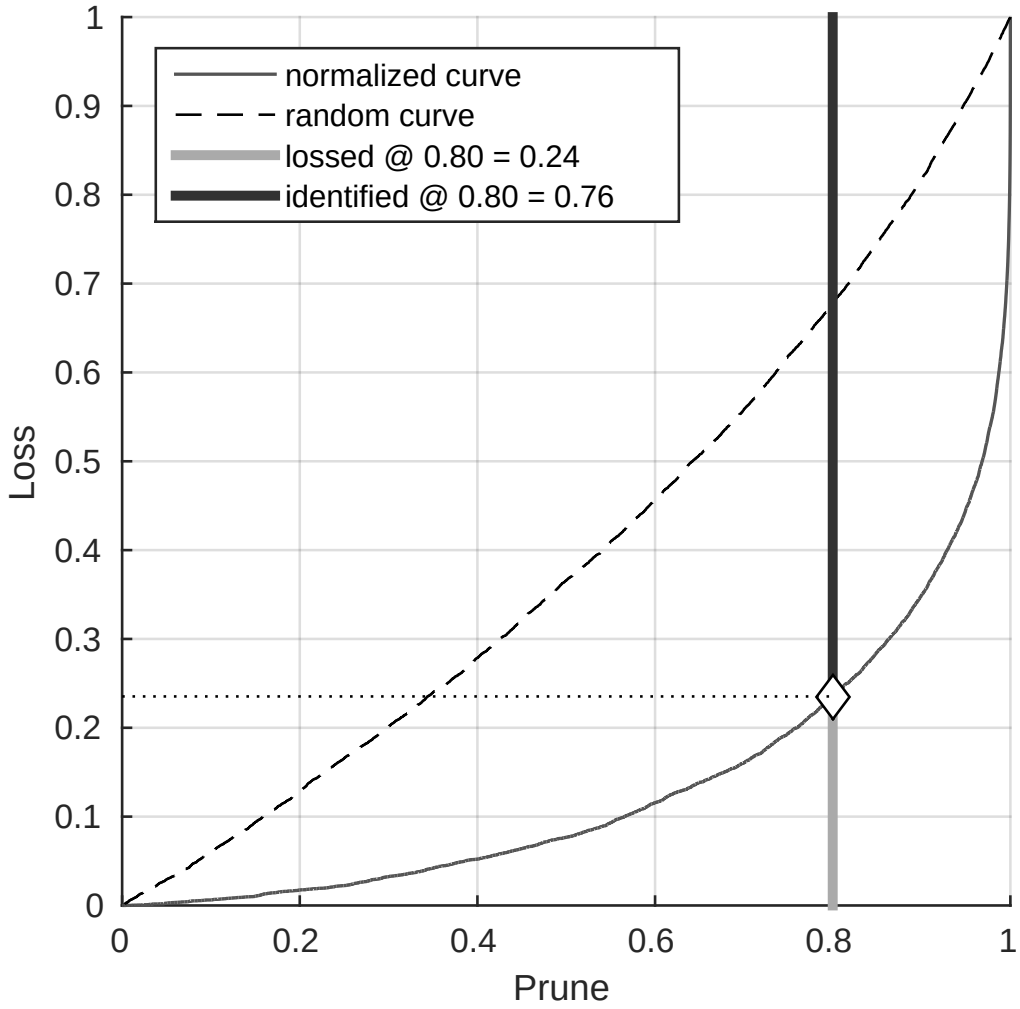
At what threshold are we guaranteed to identify all queries ?

Prune search set size



Reduce search set as much as possible to ease the work of an external expert.

Prune-Loss curve



Plot performance at all possible thresholds on a prune-loss (PL) curve.

Prune-Loss space

Prune

The prune corresponds the proportion of tracks that are rejected at a given threshold.

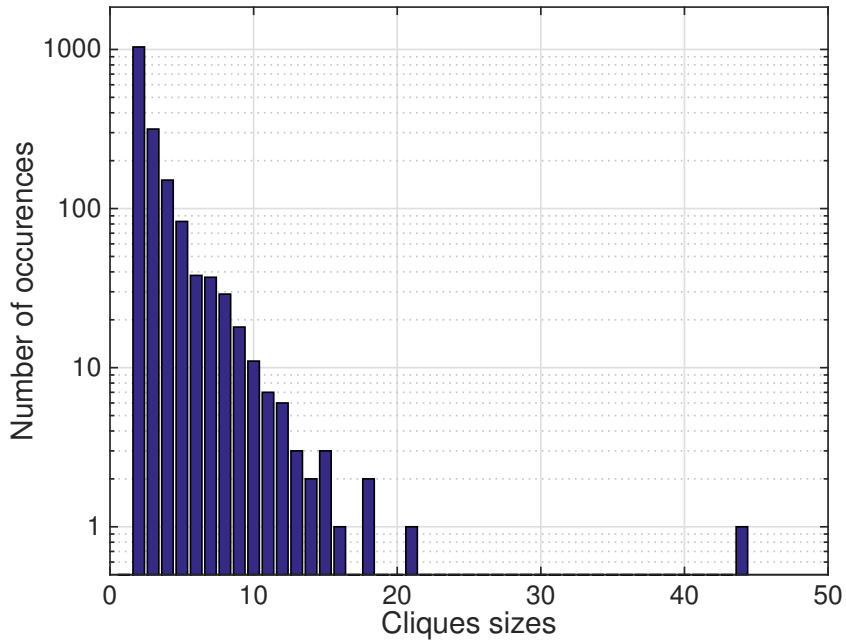
Loss

The loss corresponds the proportion of queries for which no other versions have been found at a given threshold.

$$\text{For a query, loss} = \begin{cases} 1 \text{ if no versions identified in the subset} \\ 0 \text{ if at least 1 version identified in the subset} \end{cases}$$

Normalize the loss or not ?

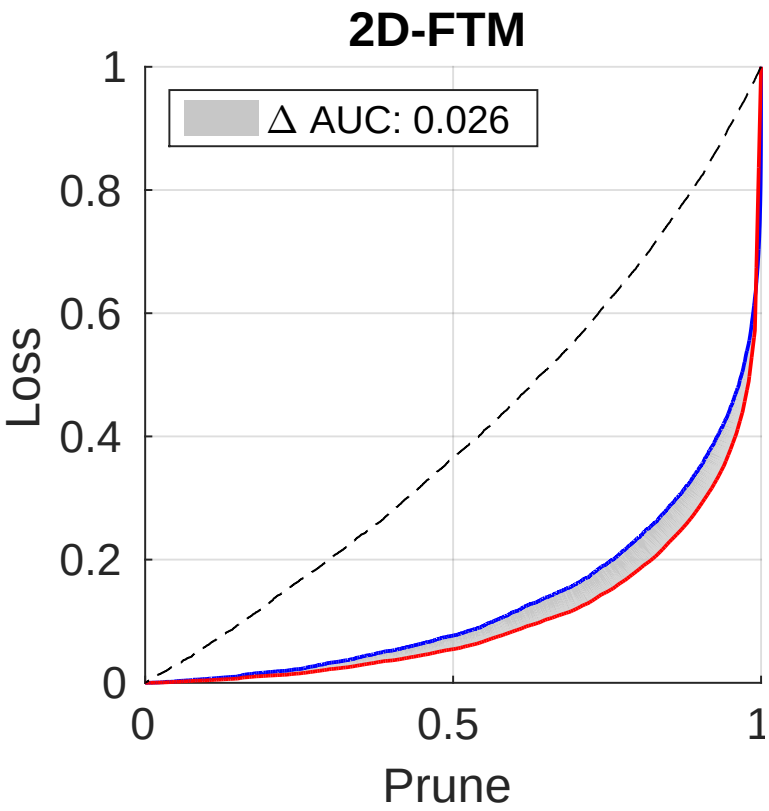
If the database is unbalanced, the cliques have different sizes:

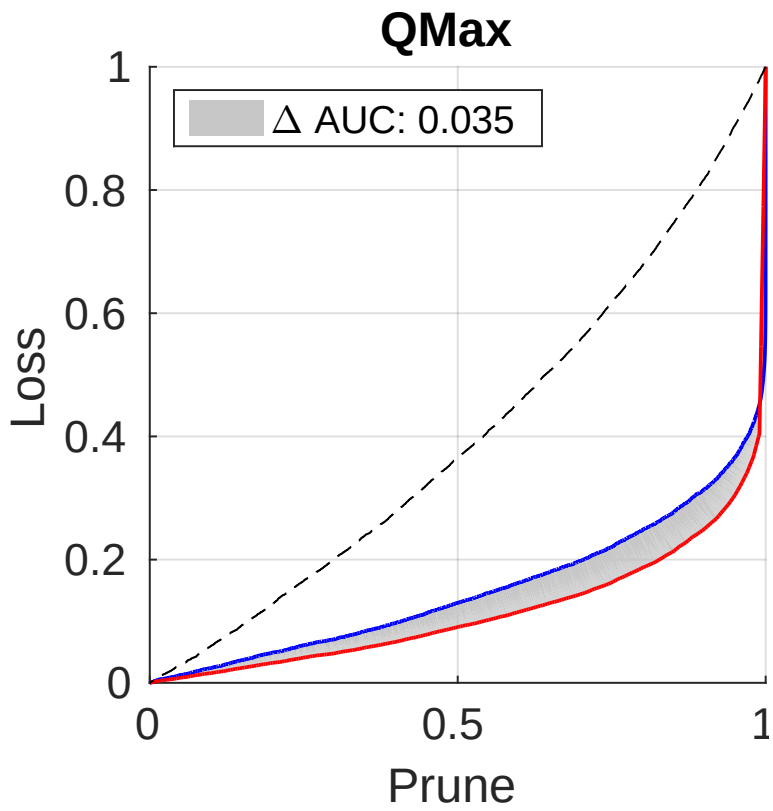


We can normalize the loss for each clique to simulate a canonical database.

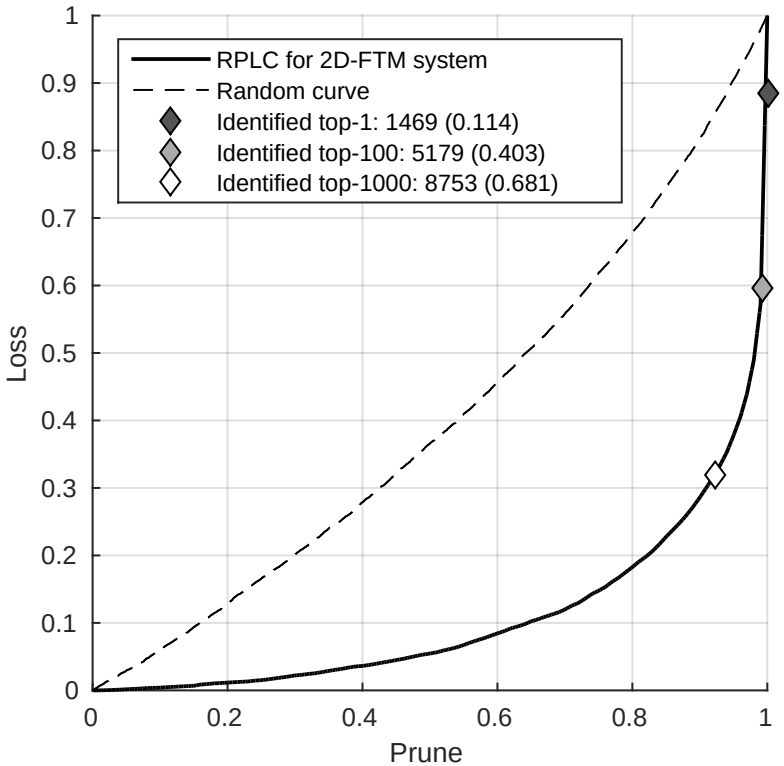
Performance on an unbalanced database

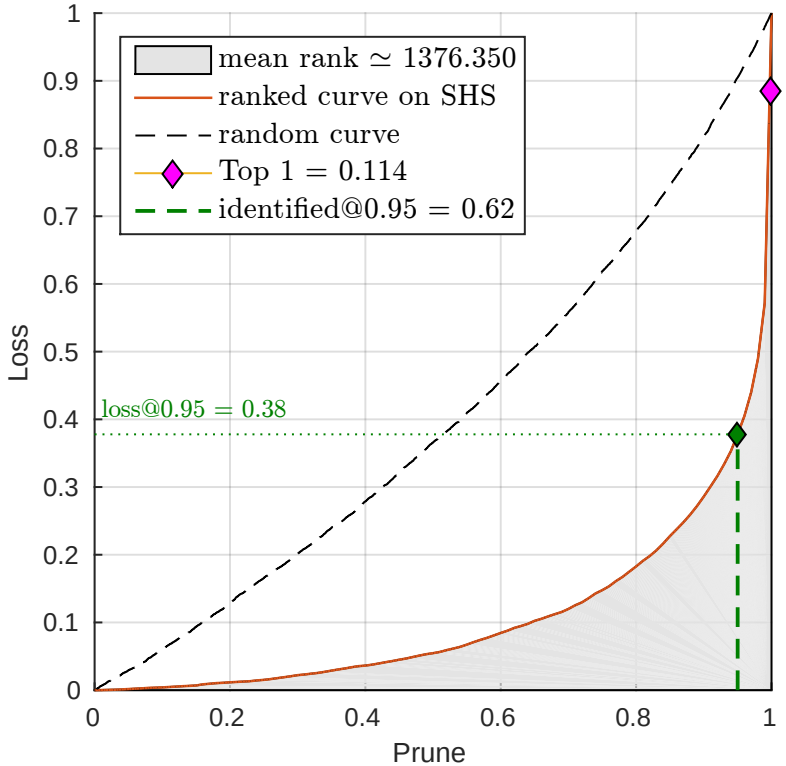
No normalization needed for specific database.





Read metrics on the PL curve



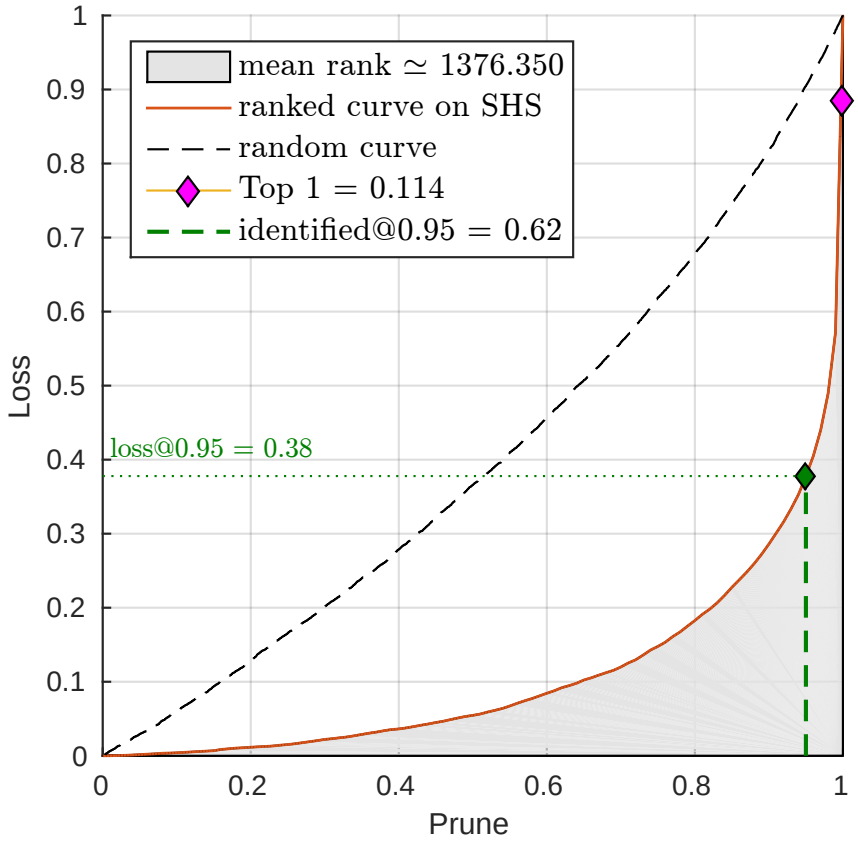


- Top-K metrics easily readable on the PL curve.
- The area under the curve corresponds to the Mean Rank metric.

Evaluation of 10 systems

Method	Features	Similarity Estimation
2D-FTM	Chroma features	Cosine Similarity
QMax	Chroma features	Alignment algorithm
SiMPle	Chroma features	L2 Euclidean
Timbre	MFCC Features	Alignment algorithm
XCorr	Chroma features	2D Cross correlation
Beats	Number of beats	Random Forests
AVG Chroma	Average chroma vector	Random Forests
Cluster	Histogram of codewords	Cosine Similarity
Duration	Duration of the songs	Random Forests
Tempo	Tempo of the songs	Random Forests

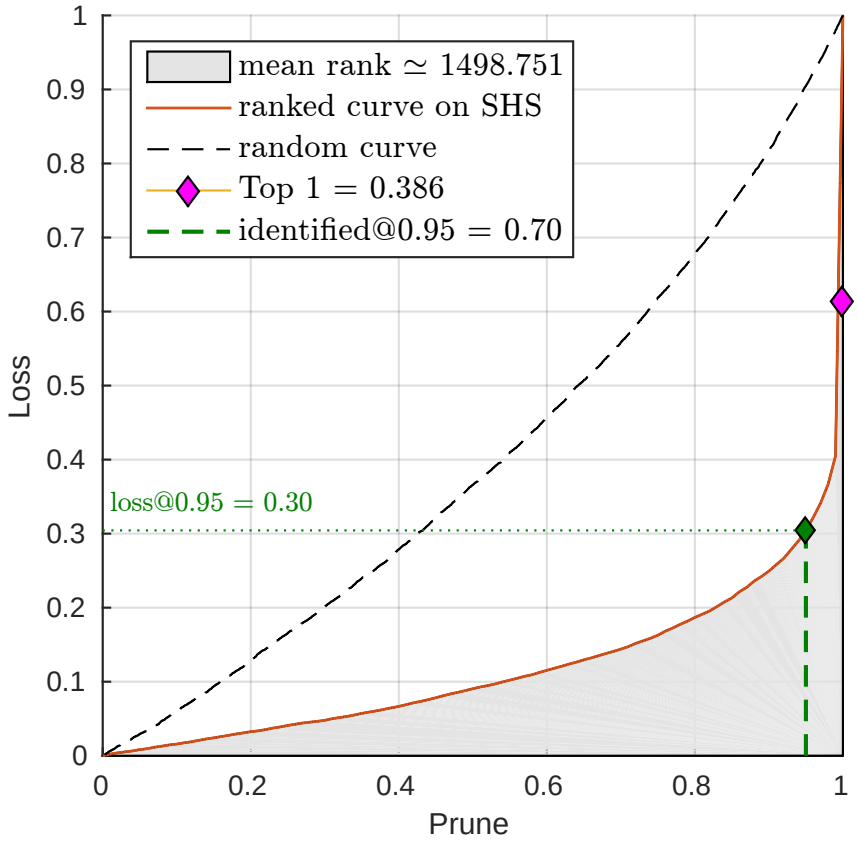
2D-FTM system



MR	MRR	MAP	Top-1	Top-10	Top-100	ROC AUC
1,359	0.15	0.08	1,469	2,869	5,179	0.769

Performance metrics on SHSD (12,856 tracks) for the 2D-FTM system

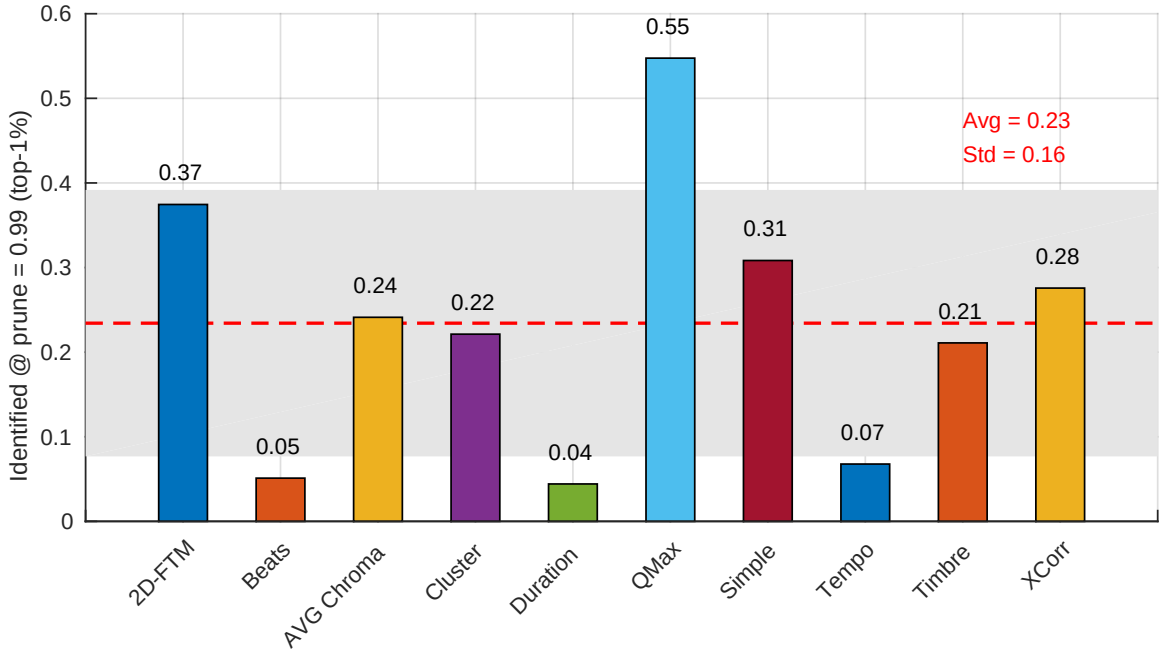
QMax system



MR	MRR	MAP	Top-1	Top-10	Top-100	ROC AUC
1,466	0.42	0.24	4,965	6,188	7,505	0.740

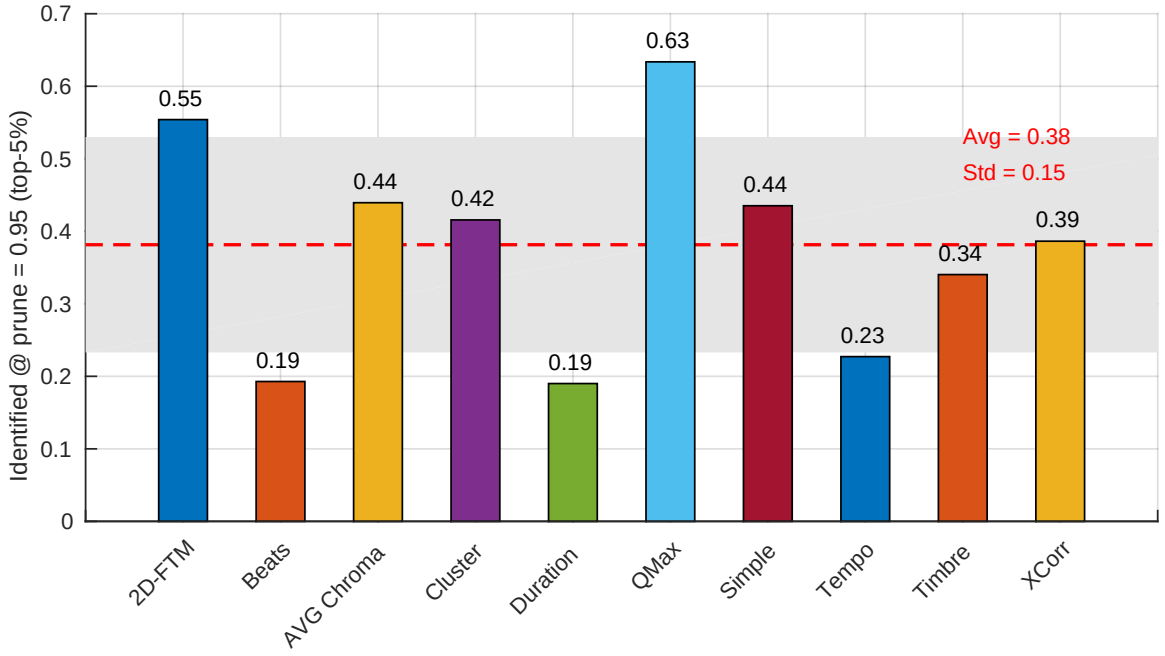
Performance metrics on SHS Train Set (12,856 tracks) for the QMax system

Comparative analysis - prune 99%



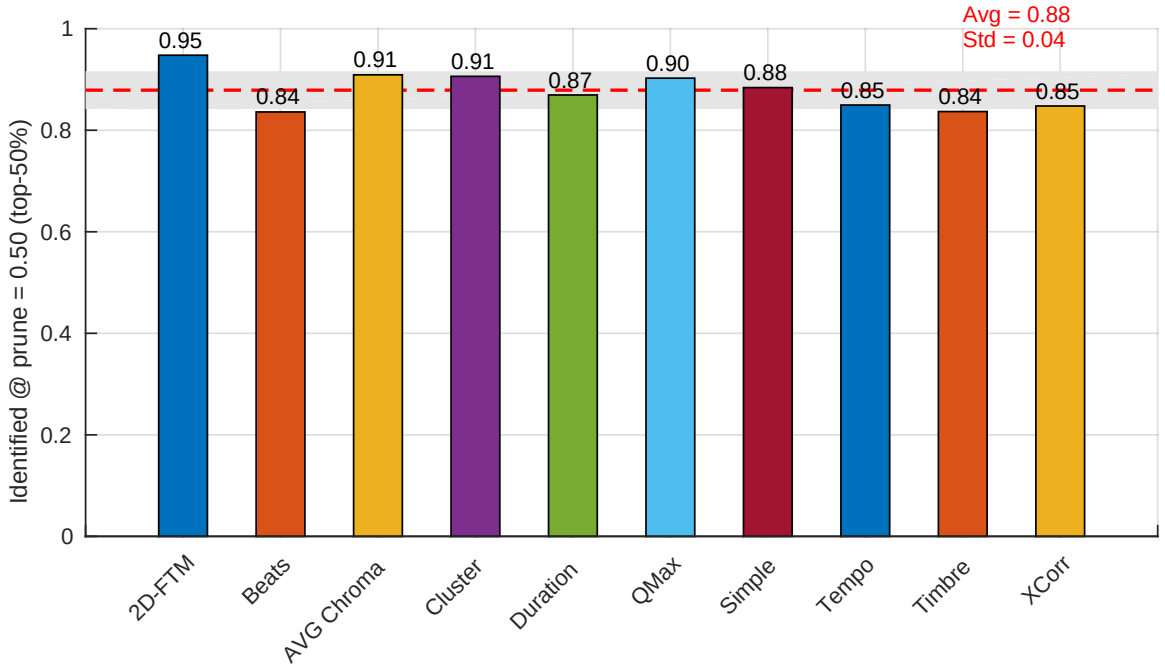
Proportion of identified tracks when the database is pruned by 99%.

Comparative analysis - prune 95%



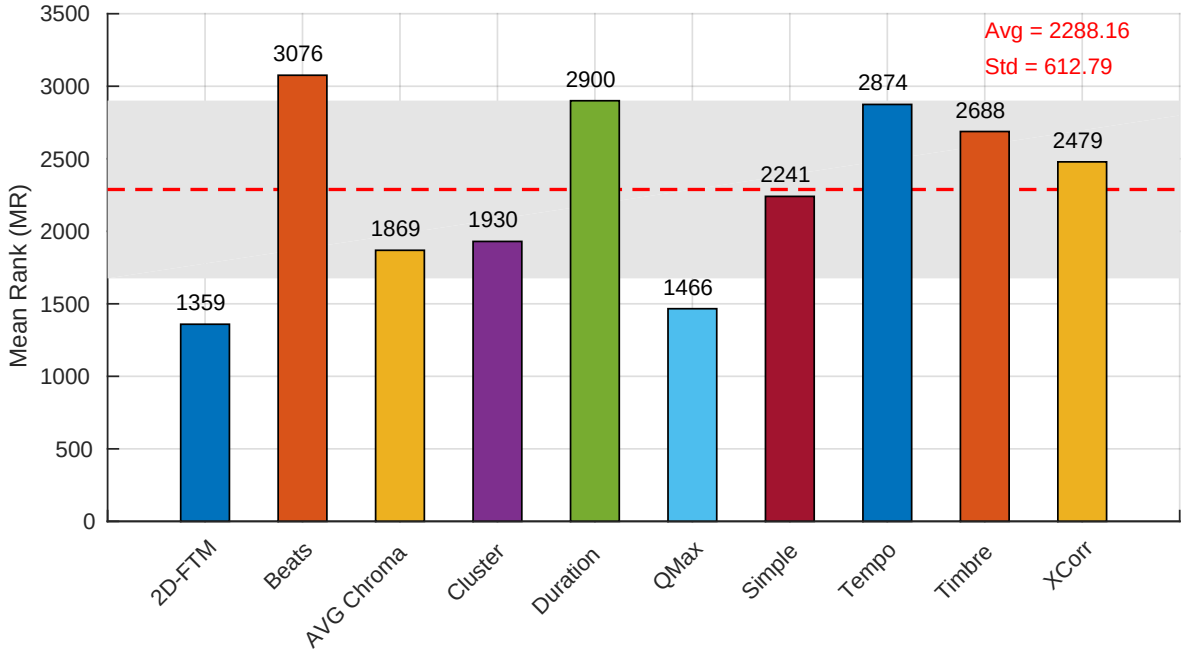
Proportion of identified tracks when the database is pruned by 95%.

Comparative analysis - prune 50%



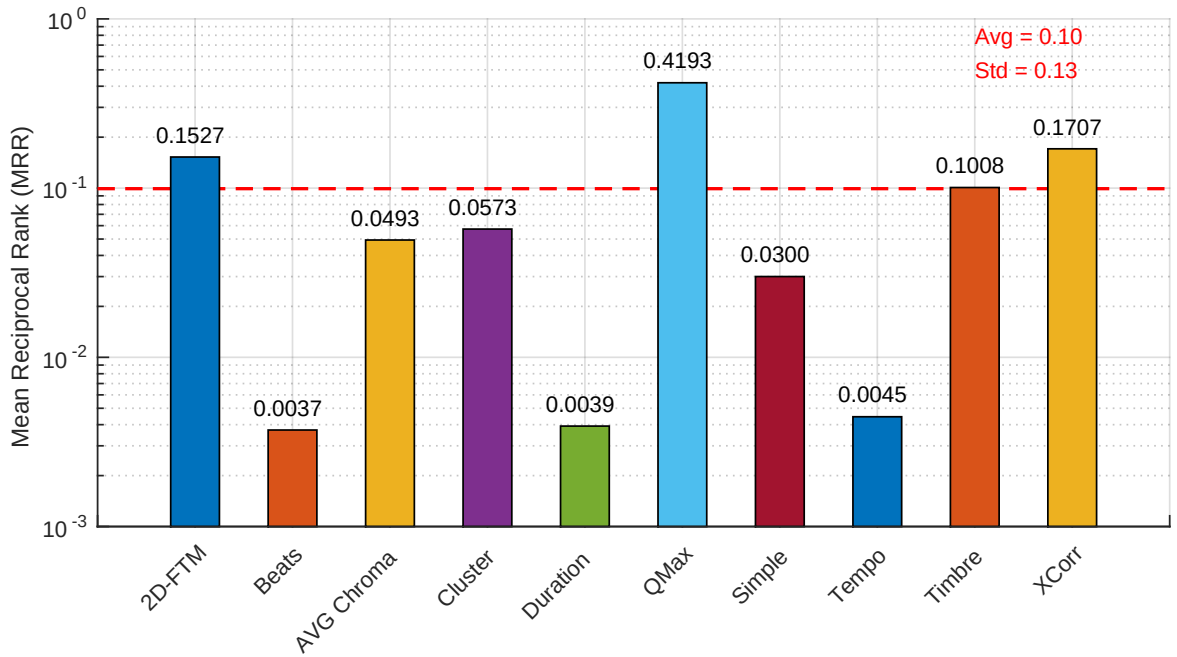
Proportion of identified tracks when the database is pruned by 50%.

Comparative analysis - Mean Rank (MR)



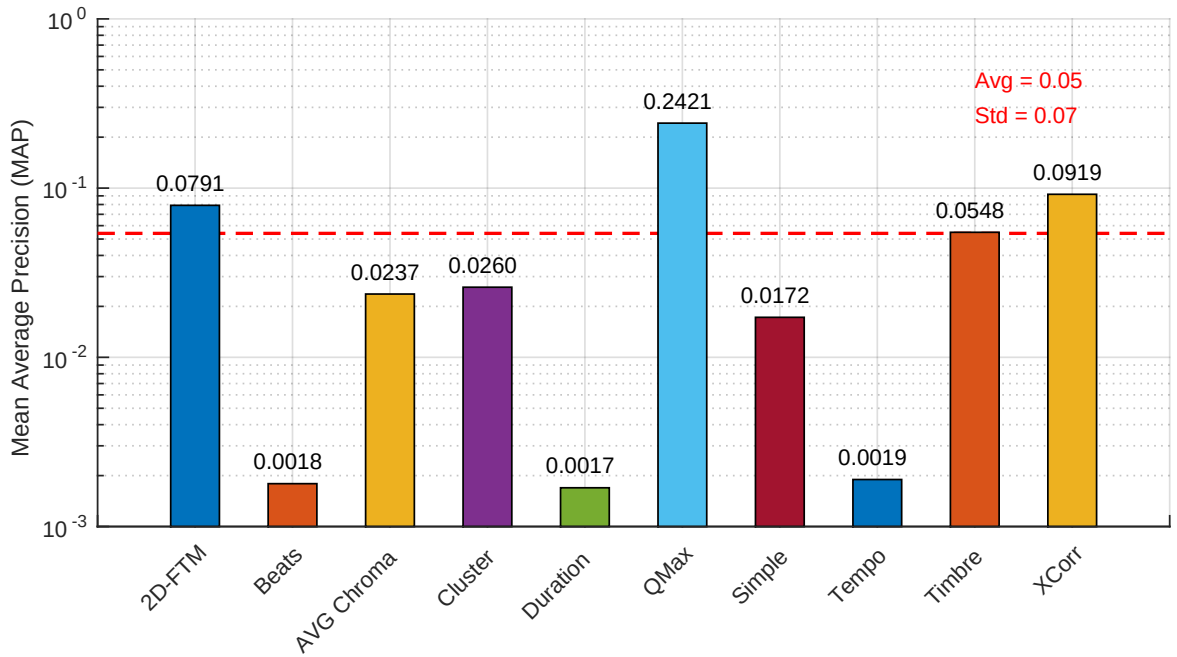
Mean Rank of the first identified match for each system. The lower the MR is, the better it is.

Comparative analysis - Mean Reciprocal Rank (MRR)



Mean Reciprocal Rank score for each system. The higher the MR is, the more queries have 1 version identified close to the top of the ranking.

Comparative analysis - Mean Average Precision (MAP)



Mean Average Precision for each system. The higher the MAP is, the more versions are identified close to the top of the ranking.

Observations

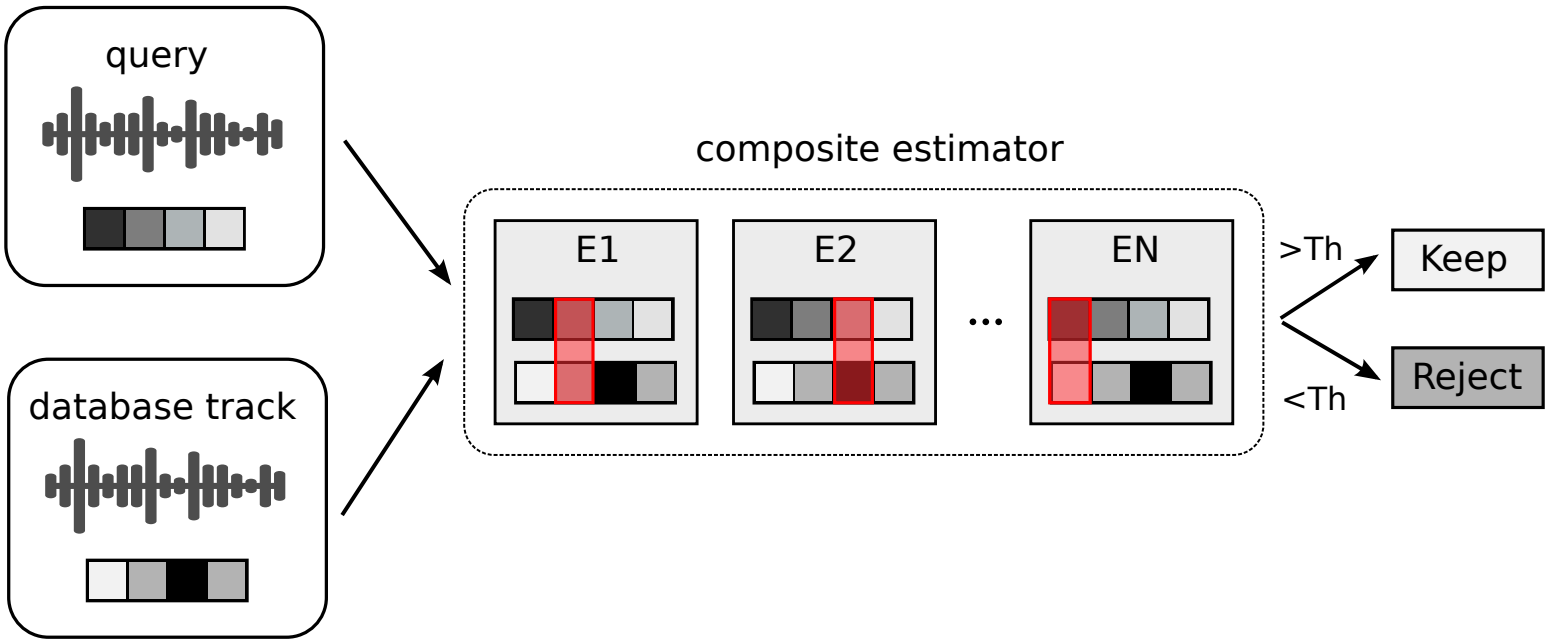
One straightforward conclusion: No system is usable for a commercial system.

- QMax outperforms all other methods.
- The problem of CSI is indeed unsolved.
- Systems based on simple features seem to perform better than random.
- When pruning at 50%, all systems perform similarly, with a good identification rate.

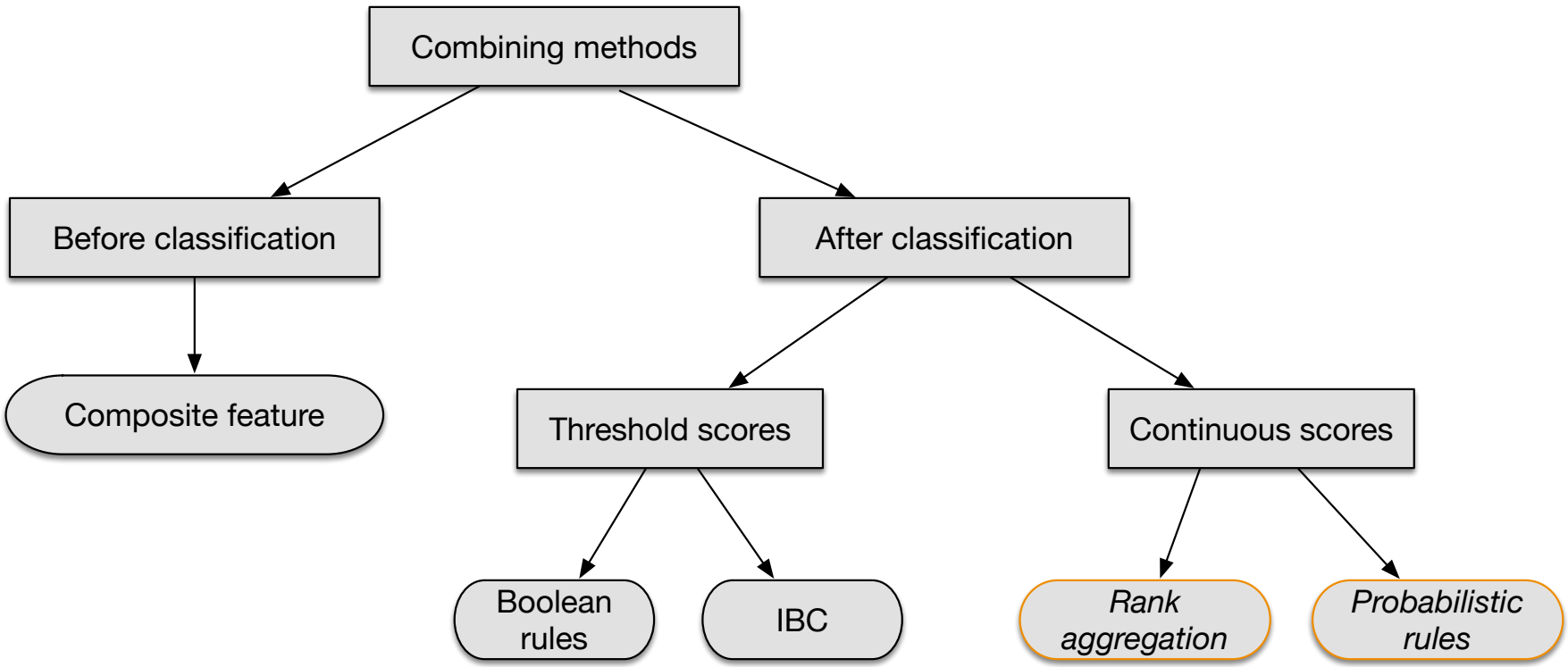
How can we improve the performance ?

Combining systems

Build a composite system that considers all initial systems to take advantage of multiple sources of information.



Different solutions exist to combine systems



We study two post-classification combining methods.

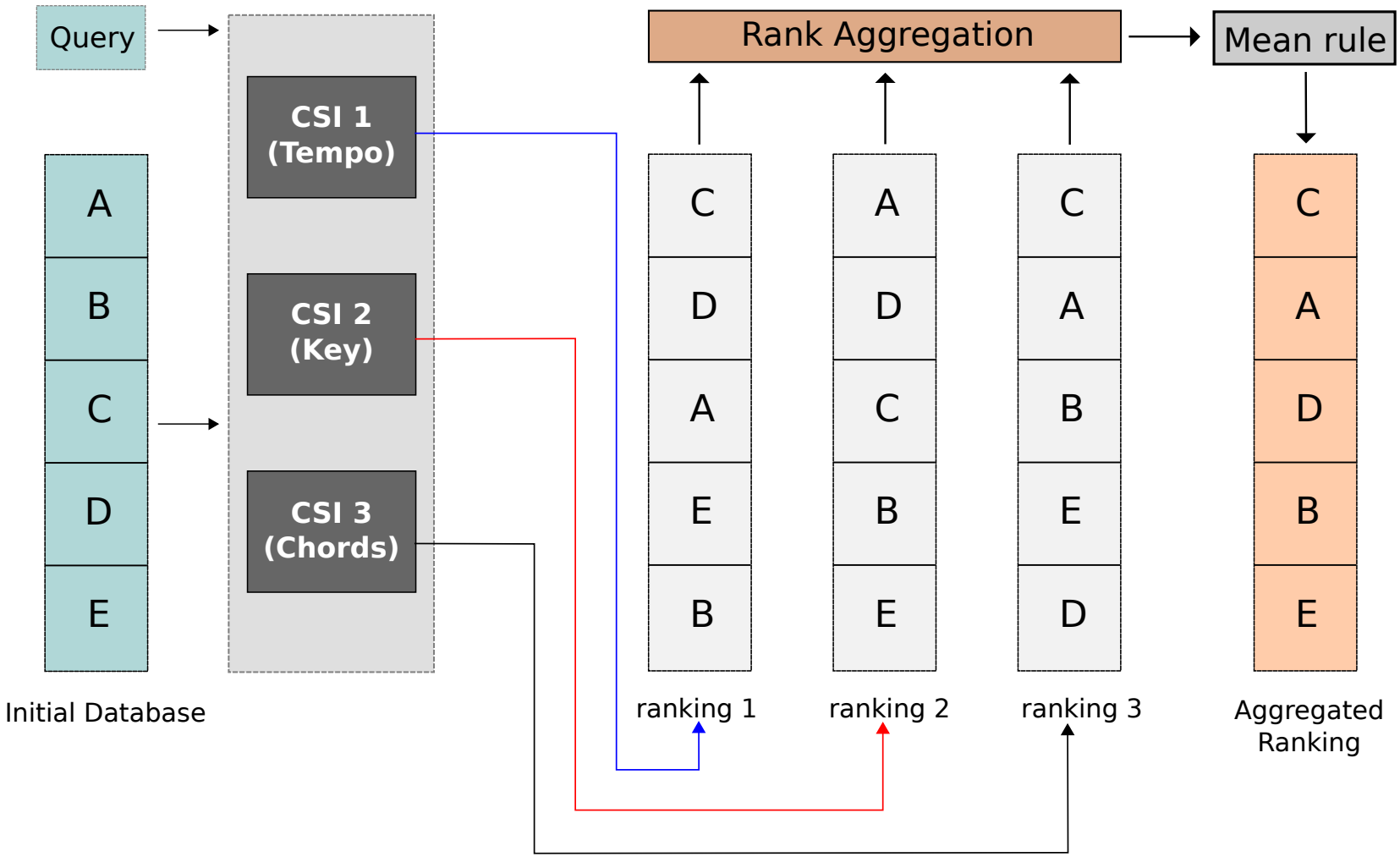
Contributions

1. Evaluation of multiple aggregation rules to combine systems

2. Evaluation of probabilistic combining rules to combine systems

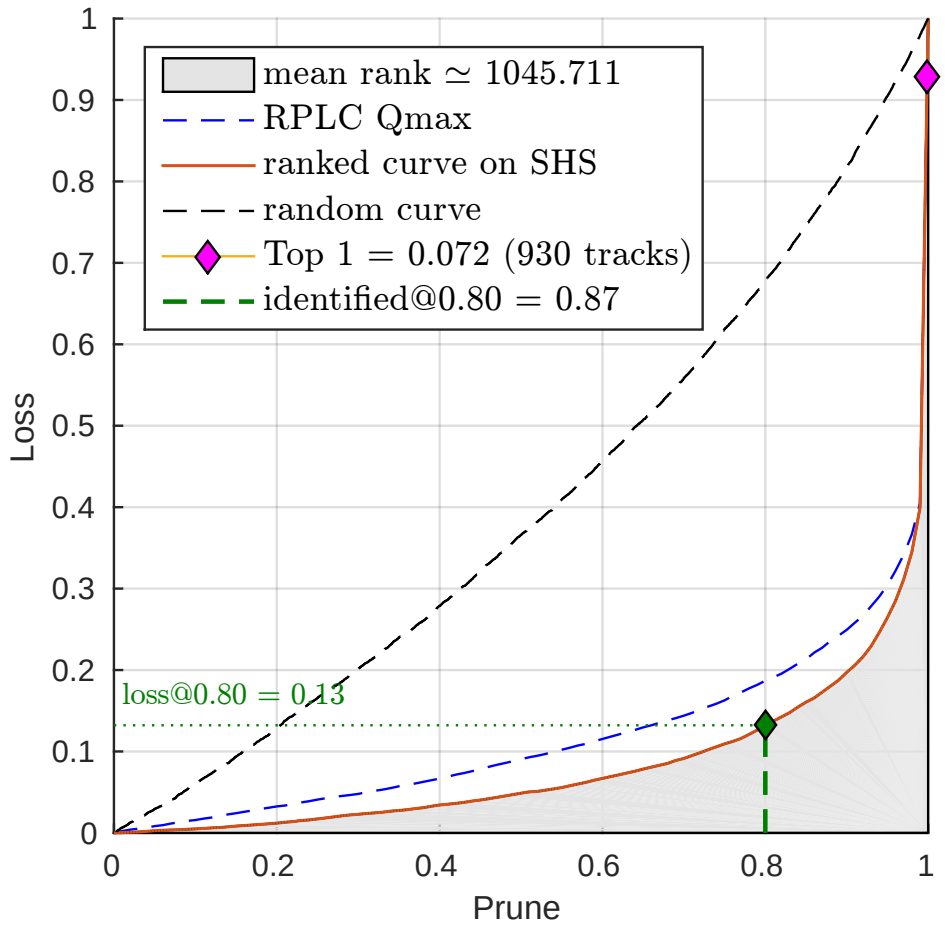
3. Implementation of the best combination in a working prototype

Rank aggregation

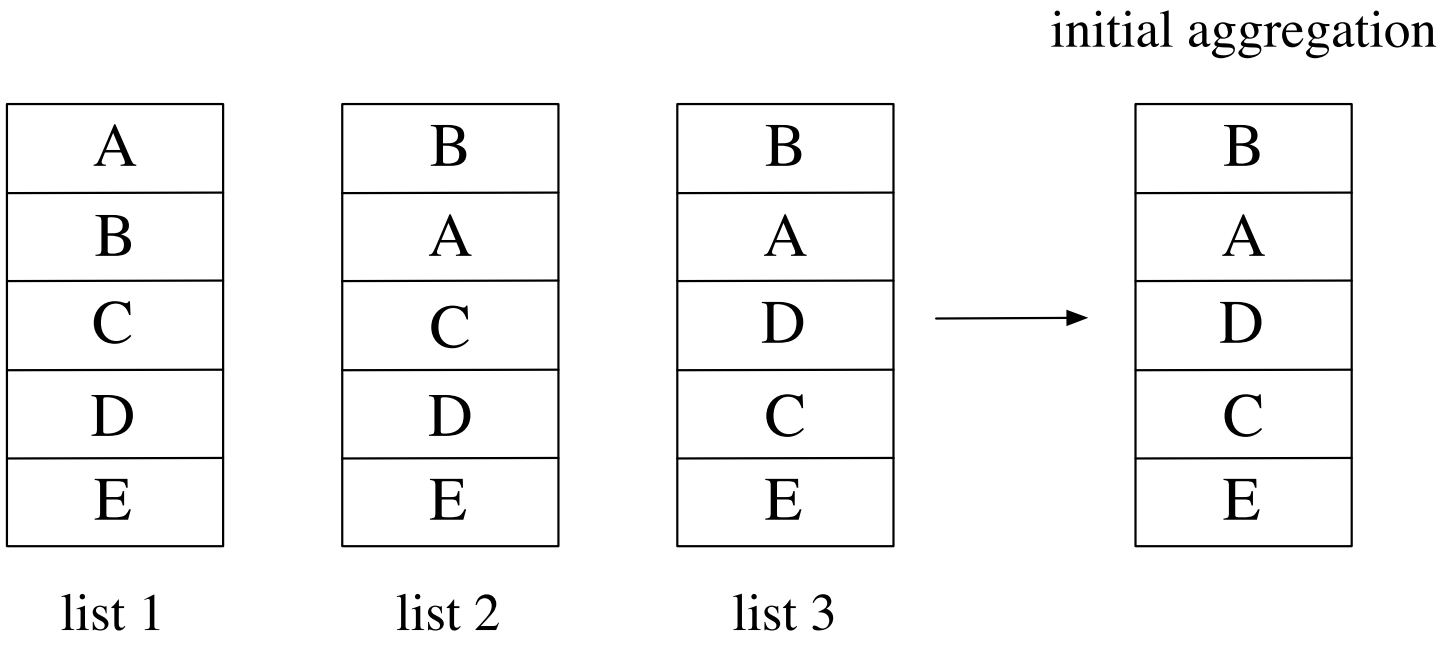


Minimum rule

Best performance is achieved with the minimum rule.

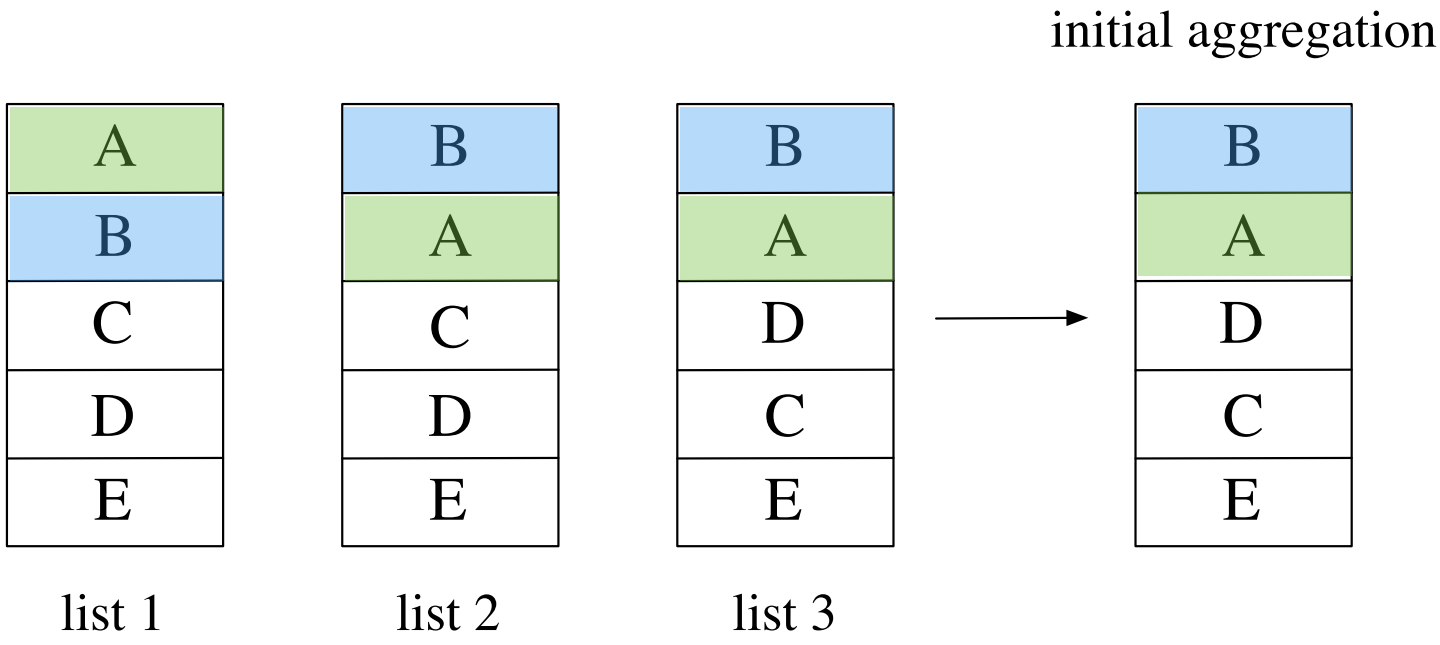


Optimal rank aggregation



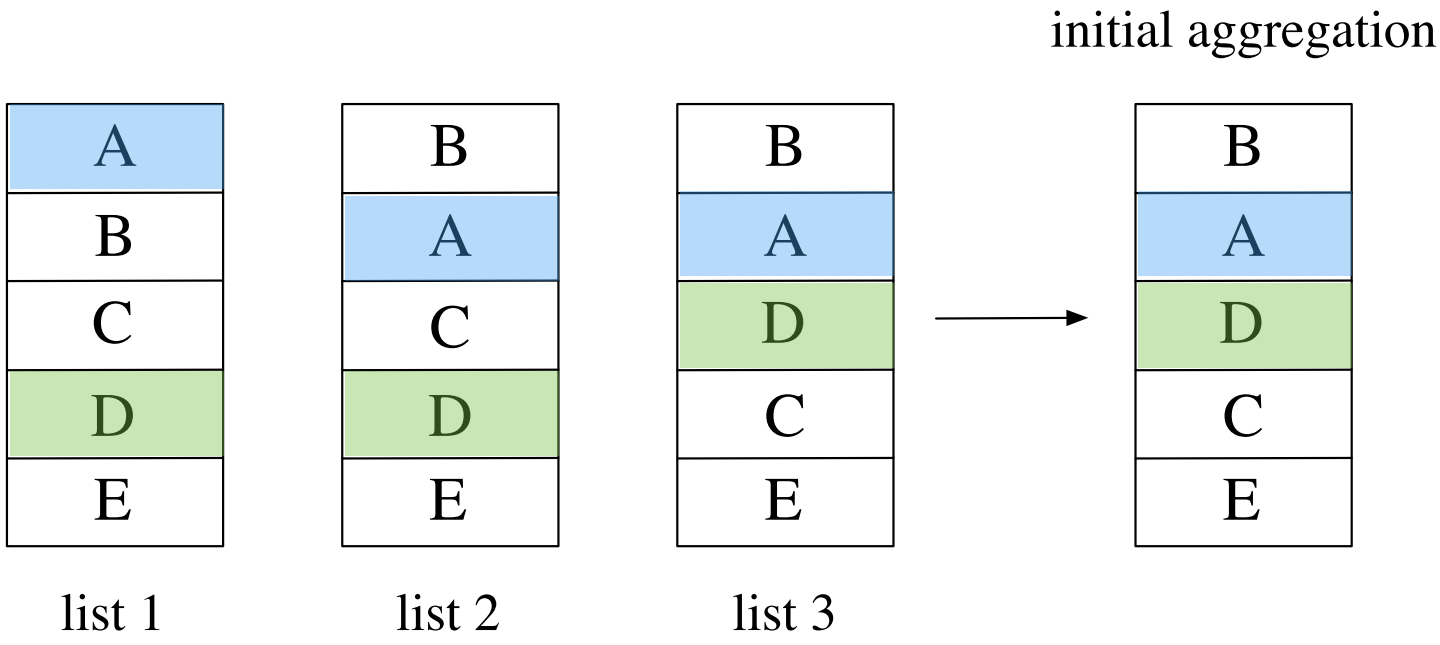
Local-Kemenization procedure improves an initial aggregation of input rankings.

Local-Kemenization



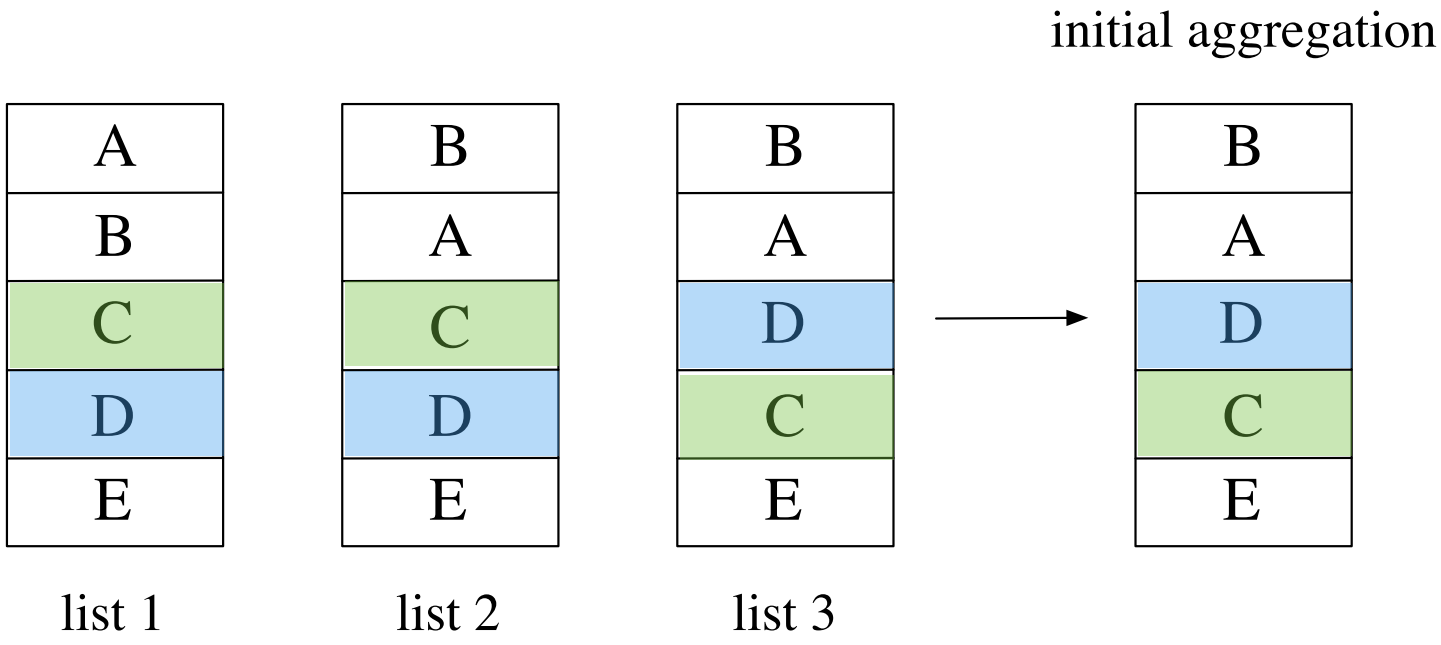
Is A above B in the majority of input lists ? No $\rightarrow$ Do nothing.

Local-Kemenization



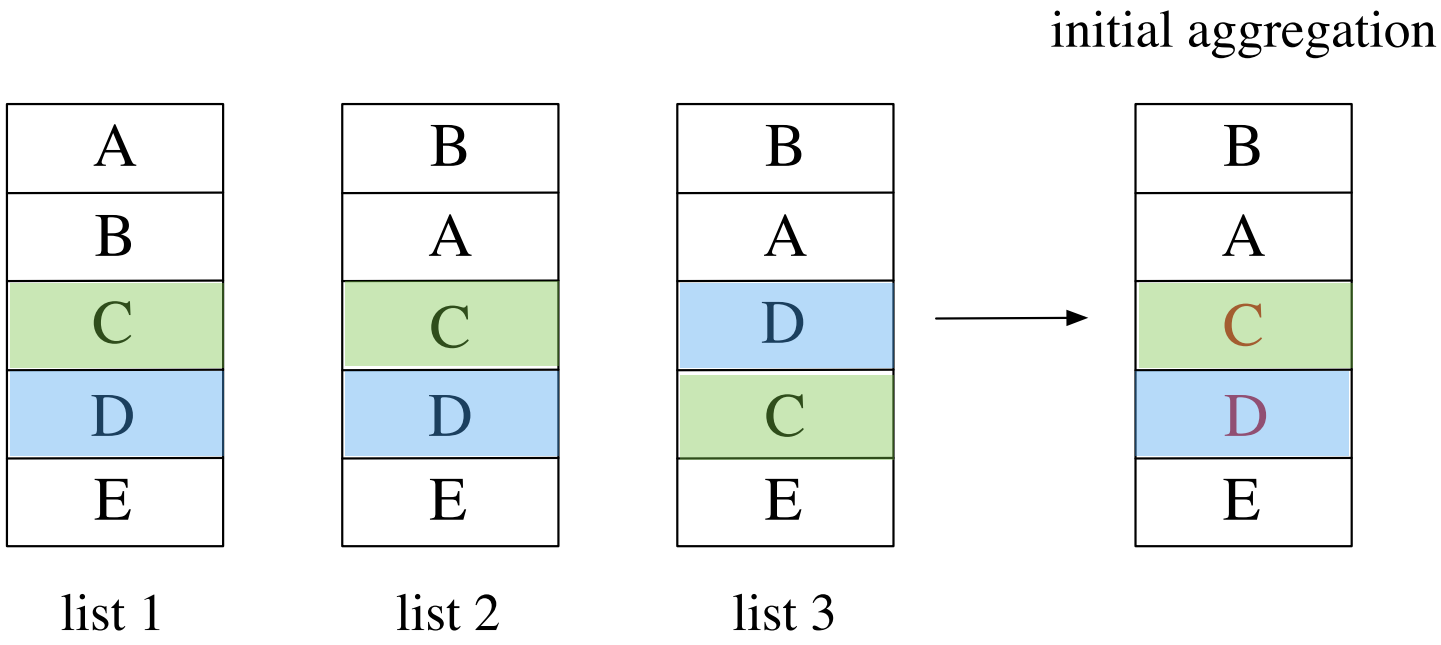
Is D above A in the majority of input lists ? No $\rightarrow$ Do nothing.

Local-Kemenization



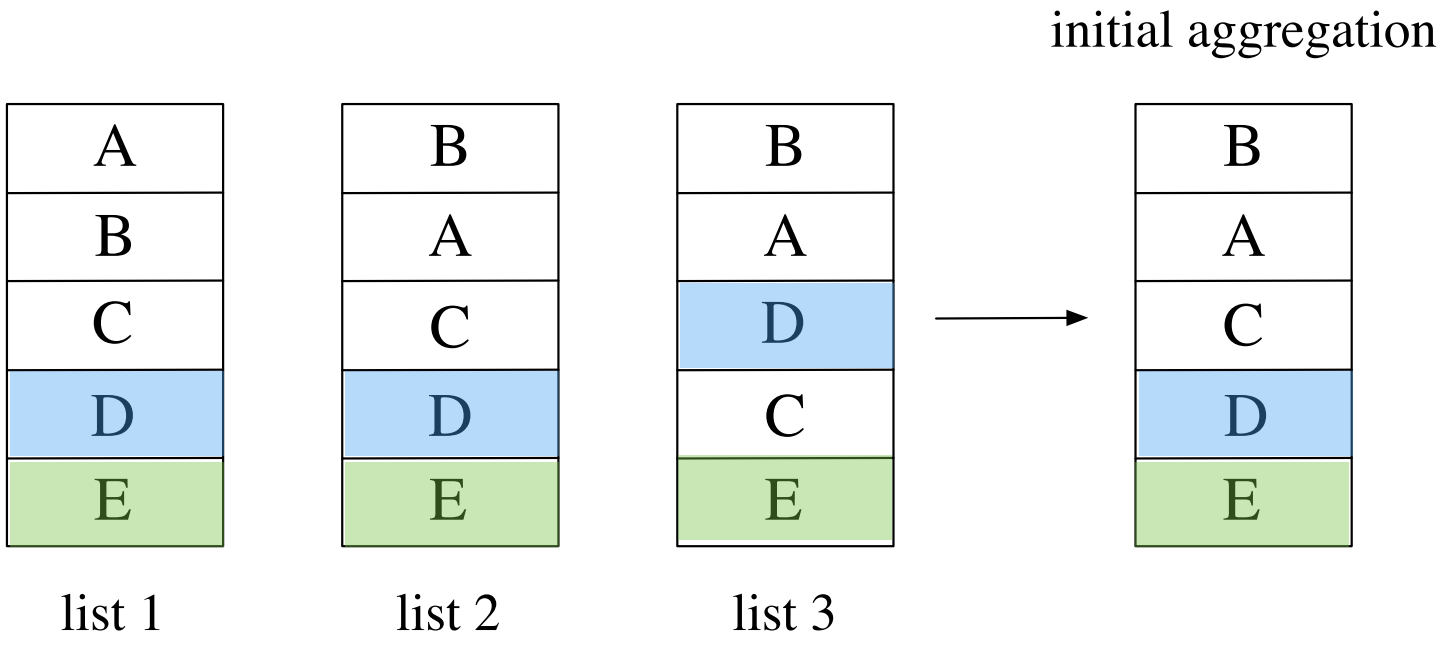
Is C above D in the majority of input lists ? Yes $\rightarrow$ Swap C and D.

Local-Kemenization



Swap C and D to make the aggregation consistent with initial lists. Continue swapping to the top until it is no longer possible.

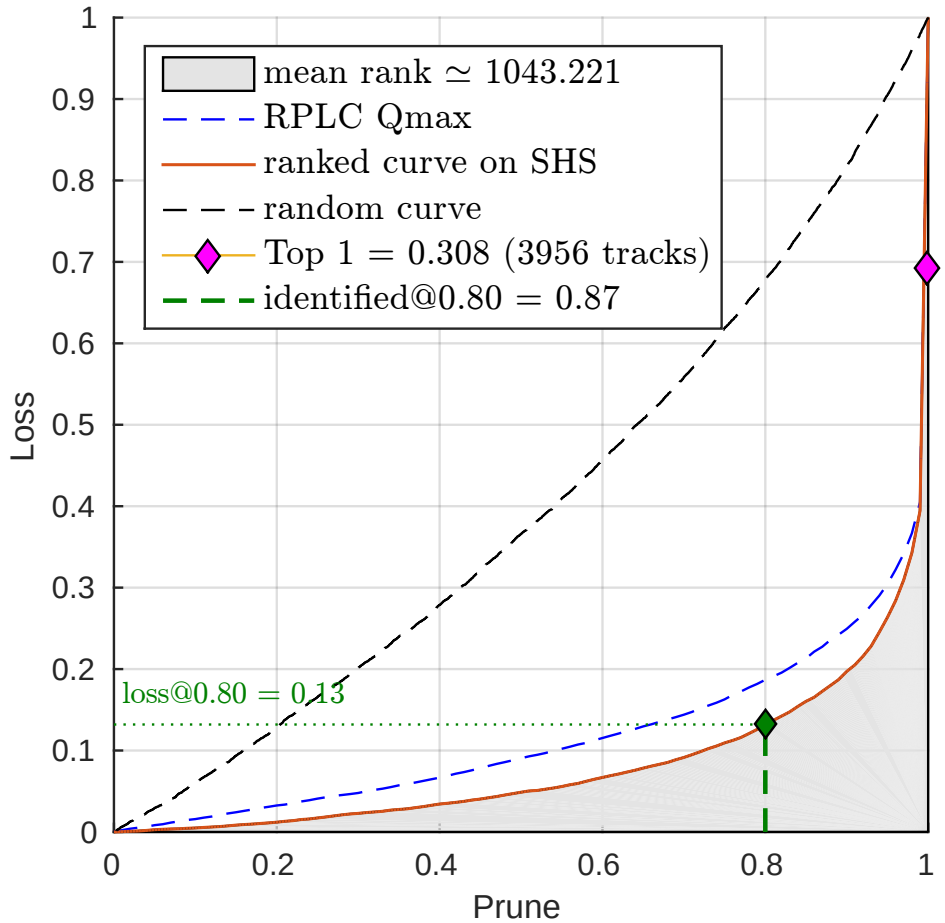
Local-Kemenization



Is E above D in the majority of input lists ? No $\rightarrow$ Do nothing.
This is the Final Aggregation.

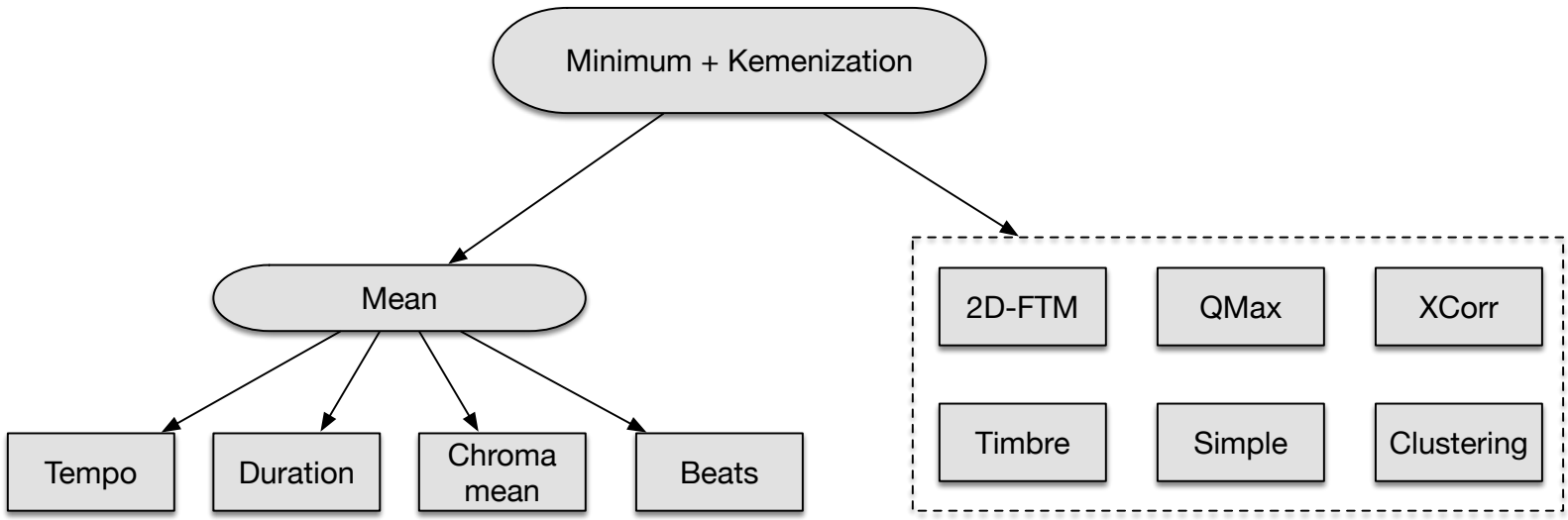
Minimum rule with Kemenization

Adding local-Kemenization to the minimum rule brings a lot of tracks to the top of the ranking.



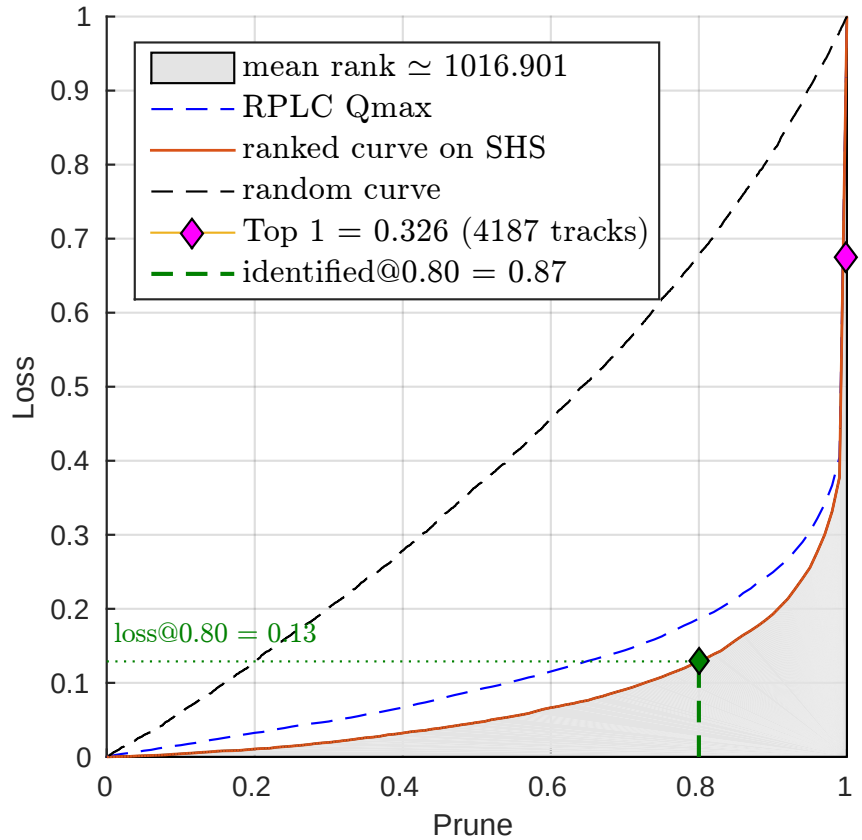
Hierarchical rank aggregation

Aggregating a subset of systems and use the resulting combination in an upper-level aggregation.



This corresponds to attributing different weights to the systems.

Performance of hierarchical rank aggregation



Method	Top-1	Top-10	Top-100	MR	MRR	MAP
QMax	4,965	6,188	7,505	1,466	0.42	0.24
Hierarchical RAG	4,187	6,433	7,790	983	0.39	0.22

Improving the combination ?

Probabilistic rules can be used for combining systems.

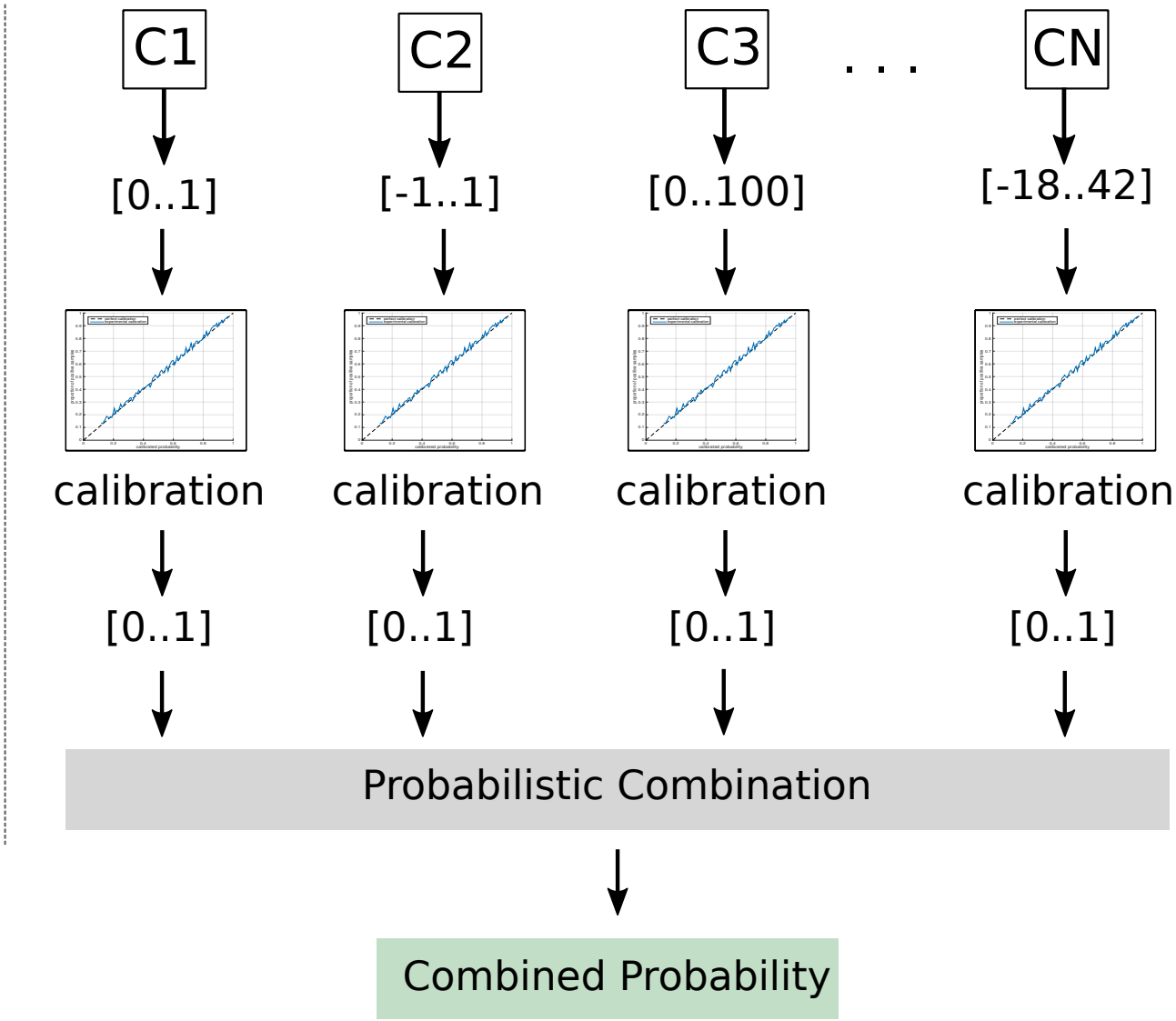
Problem: Each system returns scores on different scales.

We need to map scores to interpretable posterior probabilities.

Calibration maps a score to a posterior probability.

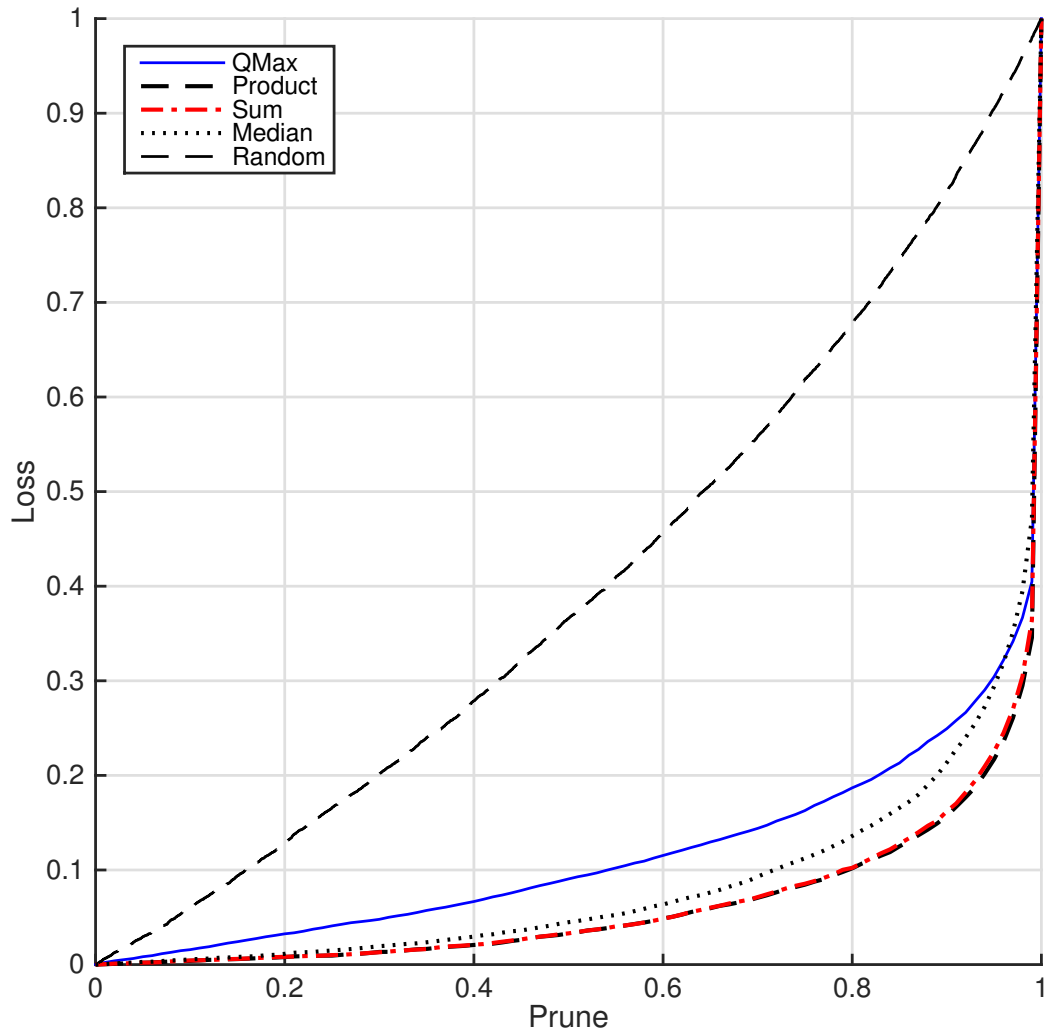
Estimate the PDF corresponding to the distribution of similar and dissimilar scores.

Calibration



Combination rules

Calibrated probabilities can be combined with 3 rules



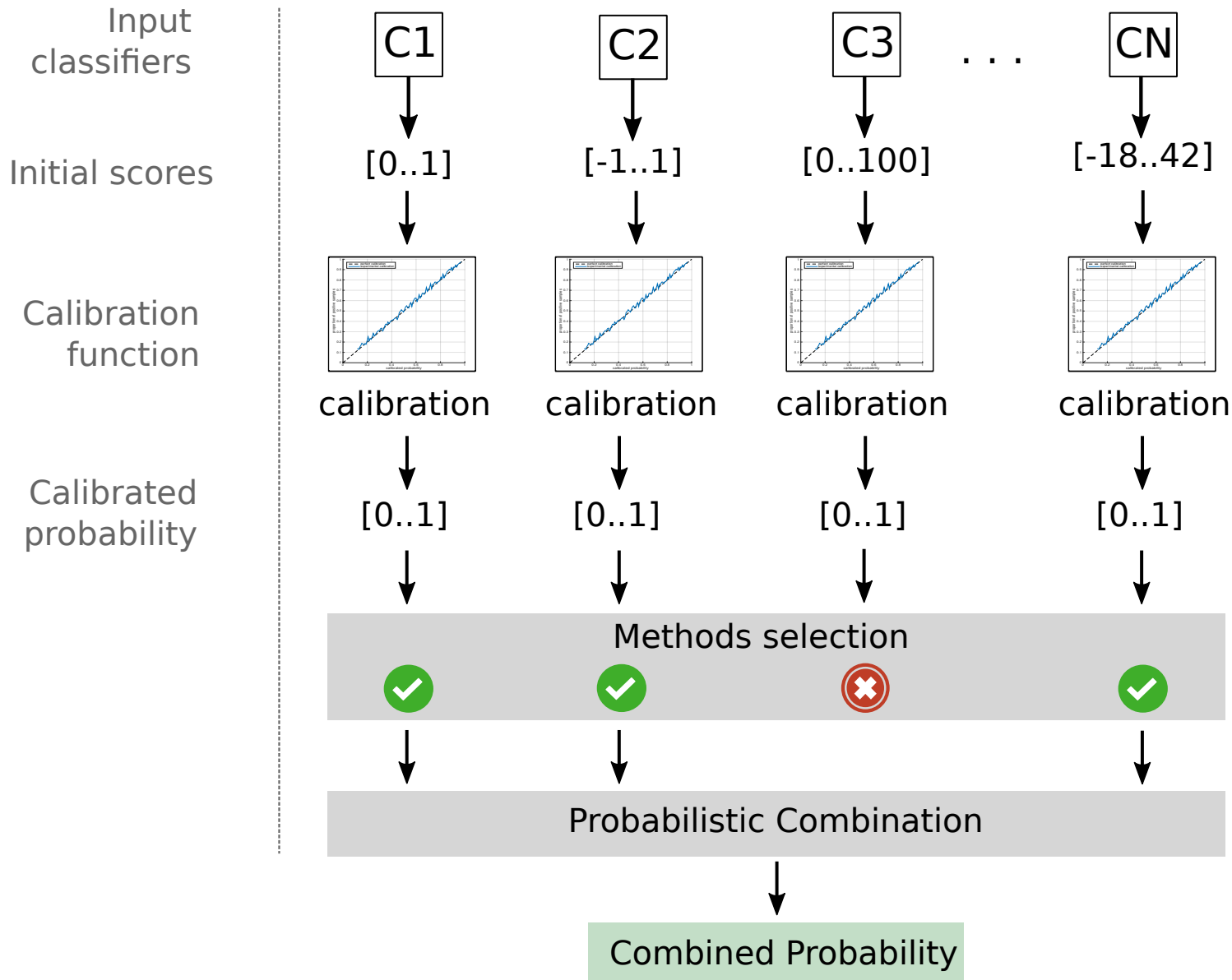
Probabilistic combination with all systems

Method	Top-1 ↑	Top-10 ↑	Top-100 ↑	MR ↓	MRR ↑	MAP ↑
QMax (Baseline)	4,965	6,188	7,505	1,466	0.42	0.24
Rank Aggregation	3,956	6,132	7,591	1,011	0.37	0.21
Product	**5,007**	**6,525**	**8,197**	**787**	**0.43**	**0.25**
Improvement	**+ 0.8 %**	**+ 5.4 %**	**+ 9.2 %**	**+ 46 %**	**+ 2.3 %**	**+ 4.2 %**
Sum	4,158	5,939	7,925	806	0.37	0.21
Median	2,598	4,189	6,473	1058	0.24	0.14

The product rule produces the best improvement with respect to all metrics. The improvement is quantified w.r.t. QMax.

Can we further improve the performance ?

Systems selection



Using a subset of systems

Best performance is achieved by removing systems from the combination, and using the product rule.

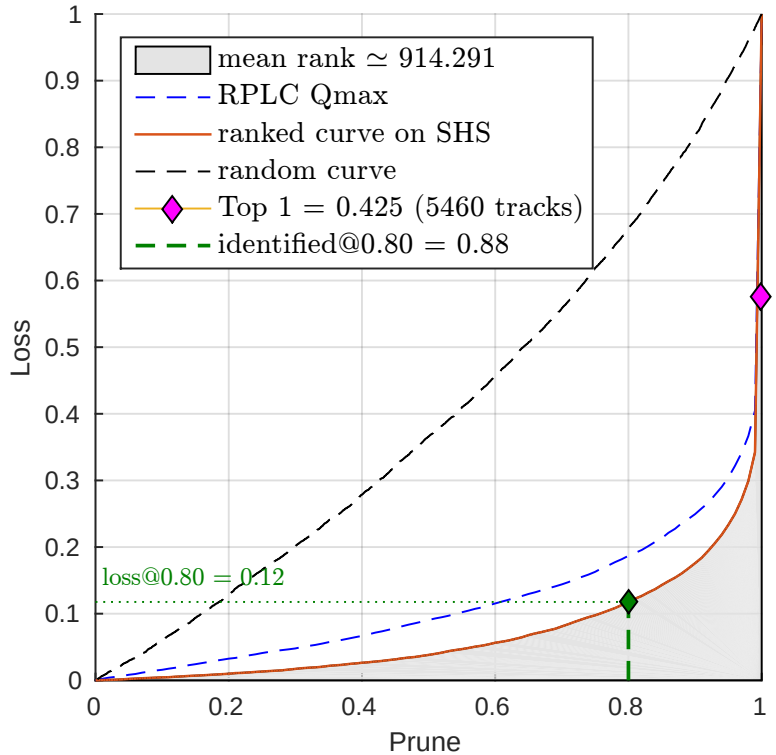
Systems dropped are:

- Chroma-Mean
- Beats
- Clustering
- XCorr

4 systems dropped means less processing time.

Performance of the subset

Method	Top-1 ↑	Top-10 ↑	Top-100 ↑	MR ↓	MRR ↑	MAP ↑
QMax (Baseline)	4,965	6,188	7,505	1,466	0.42	0.24
Rank Aggregation	3,956	6,132	7,591	1,011	0.37	0.21
Product	**5,460**	**6,816**	**8,269**	**878**	**0.46**	**0.27**
Improvement	+ 10 %	+ 10.2 %	+ 10.2 %	+ 40 %	+ 9.5 %	+ 12.5 %
Sum	4,611	6,401	8,136	885	0.41	0.23
Median	2,787	4,270	6,494	1,132	0.26	0.14

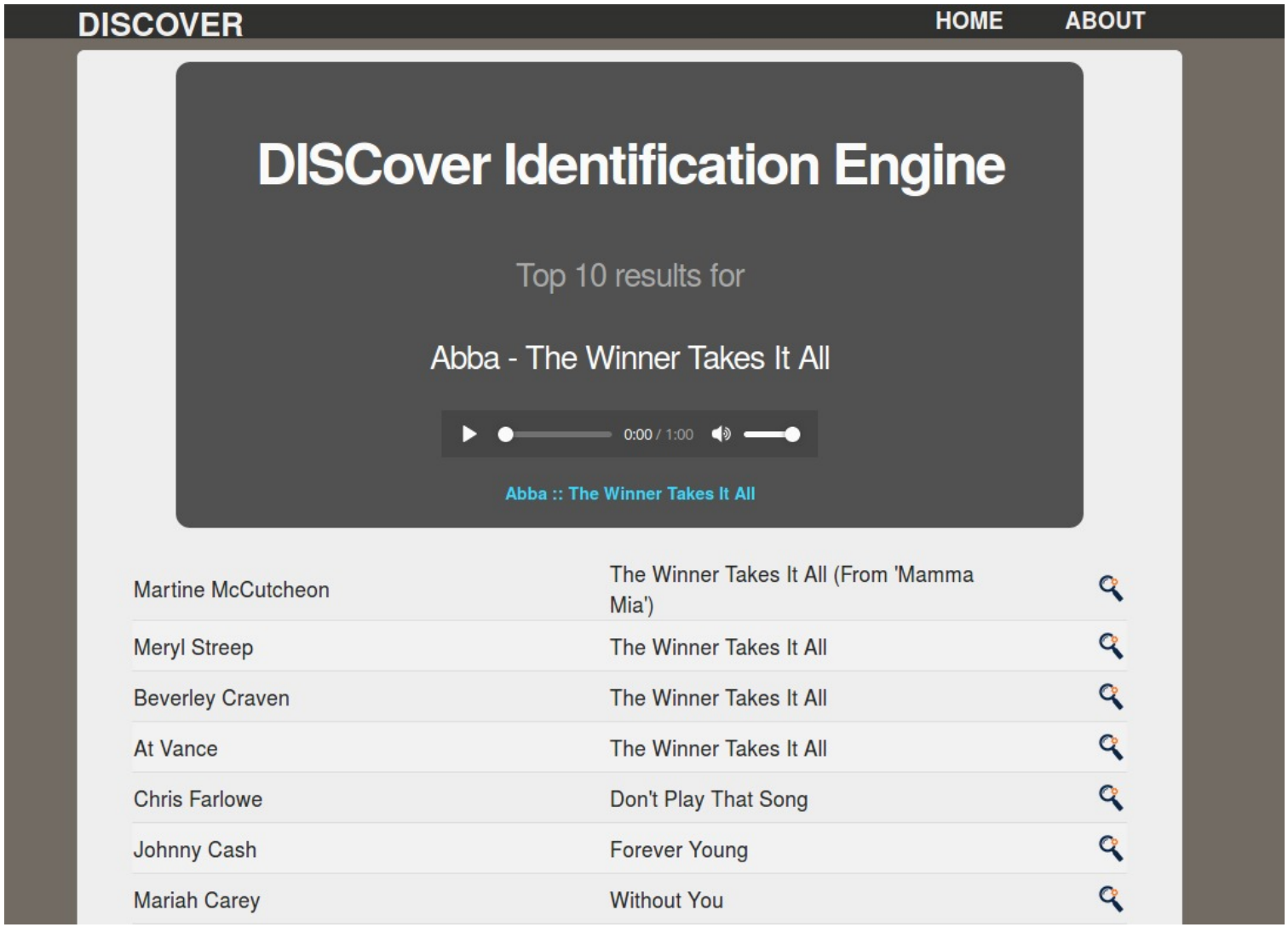


Ranking of existing systems

MR ↓		MAP ↑		MRR ↑		Identified@0.95 ↑	
method	**score**	**method**	**score**	**method**	**score**	**method**	**score**
DISCover	**878**	**DISCover**	**0.27**	**DISCover**	**0.46**	**DISCover**	**0.77**
2D-FTM	1,359	QMax	0.24	QMax	0.41	QMax	0.63
QMax	1,466	XCorr	0.09	XCorr	0.17	2D-FTM	0.55
Avg chroma	1,868	2D-FTM	0.08	2D-FTM	0.15	Avg chroma	0.44
Cluster	1,930	Timbre	0.05	Timbre	0.10	Simple	0.44
Simple	2,240	Cluster	0.026	Cluster	0.06	Cluster	0.42
XCorr	2,478	Avg chroma	0.023	Avg chroma	0.05	XCorr	0.39
Timbre	2,688	Simple	0.02	Simple	0.03	Timbre	0.34
Tempo	2,874	Tempo	0.001	Tempo	0.004	Tempo	0.23
Duration	2,900	Duration	0.001	Duration	0.003	Duration	0.19
Beats	3,075	Beats	0.001	Beats	0.003	Beats	0.19

Our final subset is implemented in an application, DISCover, and outperforms all individual methods w.r.t. all metrics.

DISCover: A working demonstrator



Conclusions

What we did :

- An evaluation framework for comparing CSI research works
- An evaluation of 10 CSI systems on a moderately large database
- An evaluation of 2 techniques for combining systems:
  - Based on rank aggregation
  - Based on probabilistic combining rules
- Combining does improve the performance !

Perspectives

- Do we use the right features ?
- Consider using features learning methods.
- Consider using deep learning algorithms.
- Pre-filter database by musical genre, or other musical characteristics ?

THANK YOU !

Example - 2D FTM

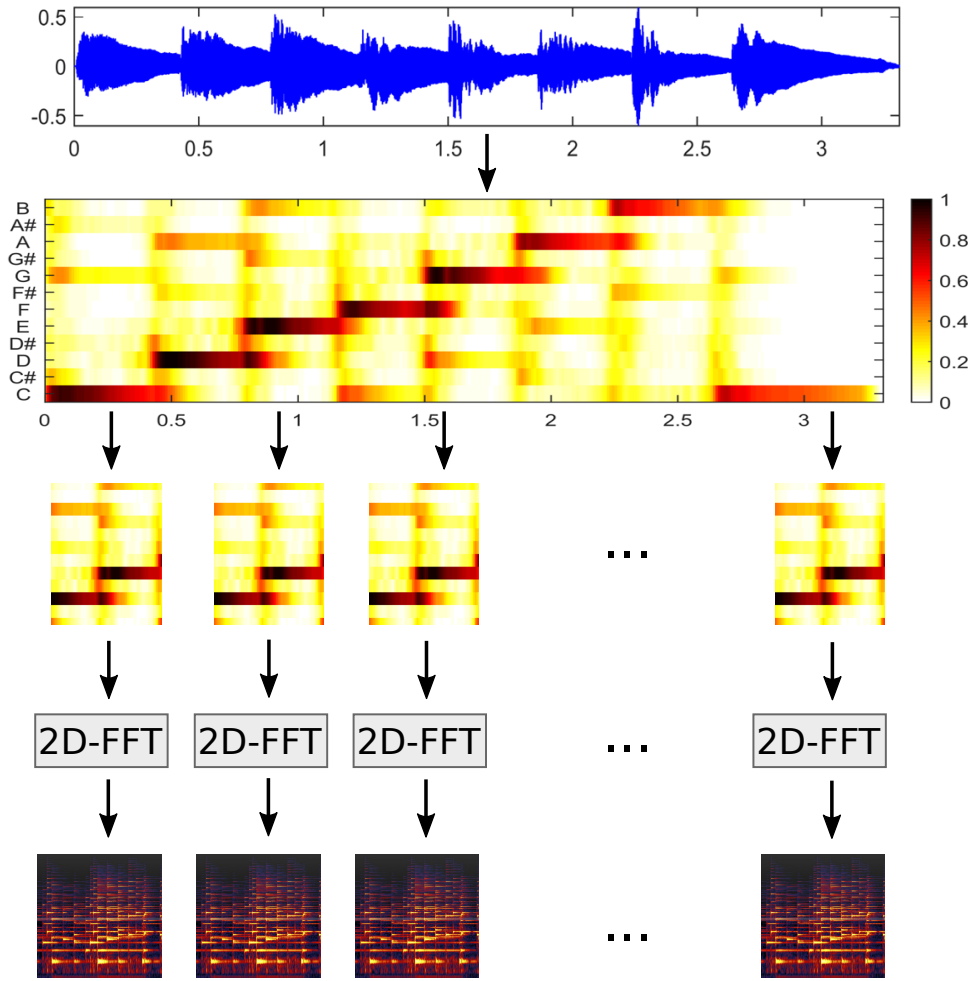
Input audio signal

Extract chroma features

Extract 75x12 chroma patches

Compute 2D-FFT for each patch

Keep magnitude coefficients

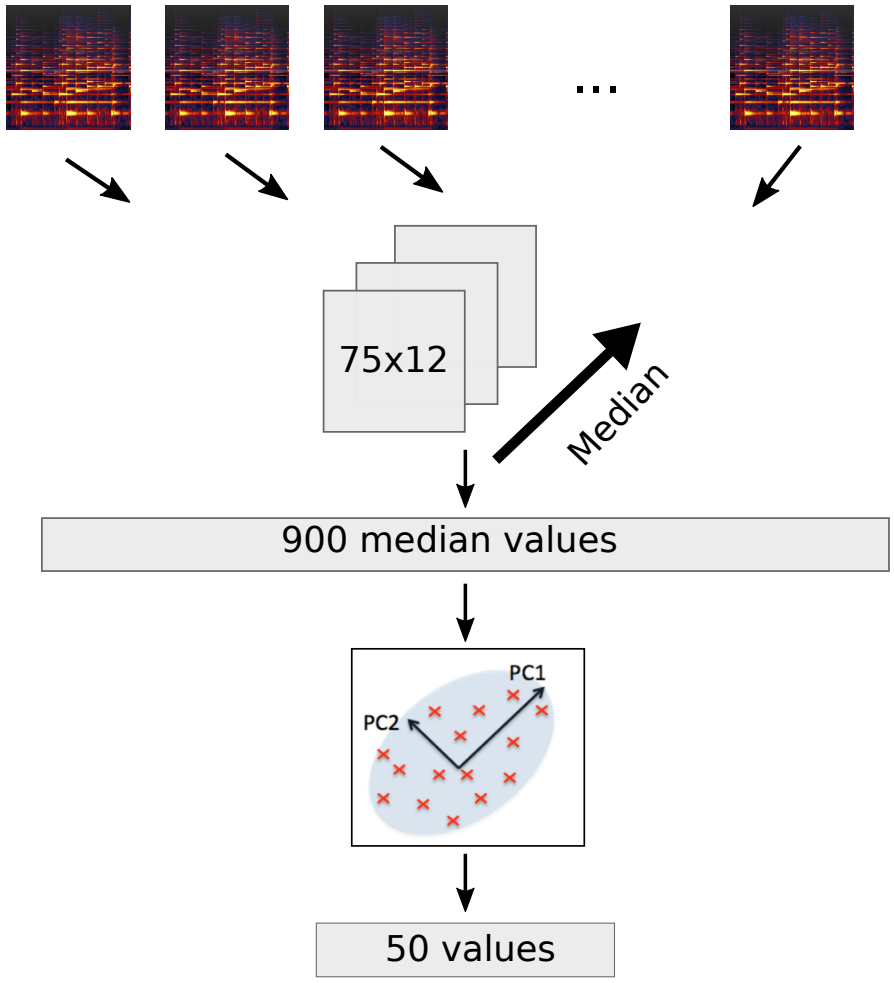


Keep magnitude coefficients

Stack patches and compute pointwise median

Apply PCA

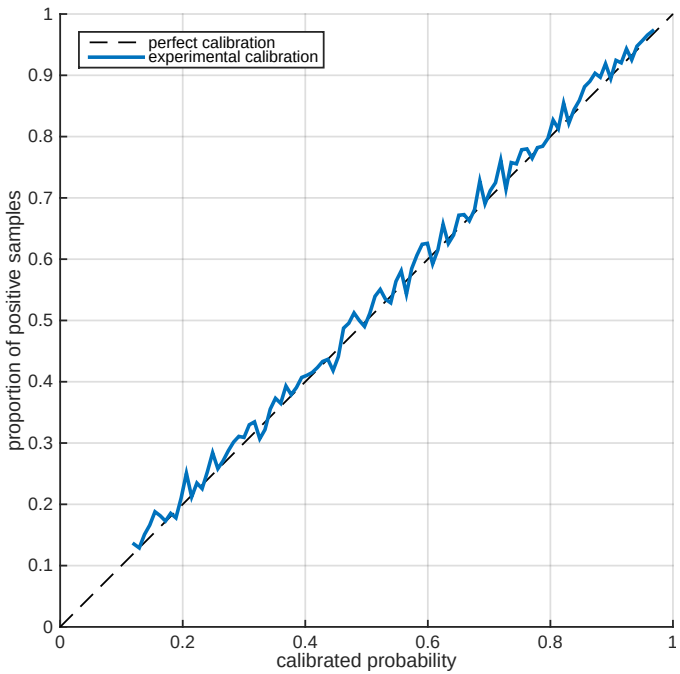
Keep first 50 PC

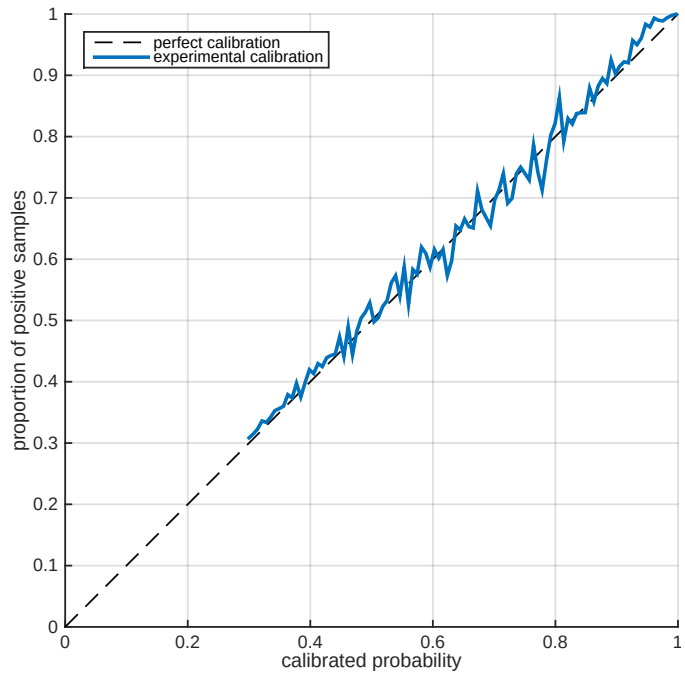


50 dimensional features vectors are compared using cosine similarity $\rightarrow$ fast comparisons !

Calibration plots

Calibration plots obtained for the 2D-FTM and QMax systems.





Well calibrated classifier: among the samples predicted with a probability of 0.8, approximately 80% of these samples belong to the positive class.