

# 1 RESULTS

**TODO** link to my github for all the code; or do it in methods, idk **TODO** put all number that are in the text into TNR

## 1.1 Implementation

This subsection is concerned with the implementation of the methods discussed in the previous section. First, the ECG data and its preparation will be discussed, followed by notes on the implementation of the SAX, MSAX, and HOT SAX methods. Lastly, the process used to analyze the results is discussed. All code used in the implementation of these methods is available upon request via email at konarski\_m@auca.kg.

**The ECG Data** The ECG data can be downloaded using the PhysioNet website at the url <https://www.physionet.org/content/mitdb/1.0.0/>, or, alternatively, using the PhysionNet-developed WFDB applications package. This package provides command line applications to work with PhysioNet data. For each of the individually numbered ECG records, 4 files exist. The .hea files contain metadata on the ECG record, including anonymized patient information and the lead names. The .dat files contain the actual ECG recording and the other two files contain additional information, including the annotations. Once the ECG recording has been downloaded, the rdsamp command is used to convert the binary ECG recording files into a more user-friendly comma separated value (CSV) file. The rdann command is then used to create a CSV file containing the annotations for each of the ECG records. Finally, the ECG recording data and the annotations can be merged into a single file by using the time stamps contained in both files. This yields full ECG recordings with added beat annotations in one file. These files are the basis of all further methods and analysis performed in this research. The author created a script in the Julia programming language that performs this process. Filtering of the ECG data is not performed. The rationale behind this is twofold. Firstly, the combination of PAA and discretization in SAX and MSAX has a smooting effect that exhibits some of the same properties as filtering. Additionally, filtering of ECGs adds many more parameters that can be modified to improve the performance of the methods, which is not desirable for this research as the methods should depend on the least possible number of parameters for simplicity. As additional support for this approach, [1] <sup>zhang2019</sup> can be considered, which successfully uses SAX in their ECG analysis without mentioning any filtering performed on the ECG data.

**SAX, MSAX, HOT SAX Implementation** The main program for this research was developed using the Julia programming language. Julia is a scientific programming language that has similarities to R, MATLAB, and Python. Julia possesses a rich ecosystem of libraries for visualization, computation, and data manipulation. For more information, visit the Julia website at <https://julialang.org/>. The following subsection will detail the steps comprising the discord discovery program.

The first step is the selection of the important parameters for the methods. The user defined parameters are:

- the sampling frequency of the ECG data to be analyzed;
- the number of PAA segments  $w$  used for SAX and MSAX;
- the alphabet size  $a$  used for SAX and MSAX;
- the subsequence length that determines HOT SAX;
- the variable  $k$  indicating how many discords should be found.

These parameters determine all actions the program performs afterward. The second step is to load a CSV file containing the ECG data and annotations into the program. Once the ECG file is loaded, it is transformed into a data frame. A data frame is a type of data structure that can hold heterogeneous data types, e.g. text and numbers. This step adds important information to the ECG data. The ECG data frame contains the parameters itemized above to enable reproduction and analysis of the results, an index range for each PAA segment so it can be located in the raw ECG, the beat annotations for each PAA segment, and empty data fields for the results of the analysis with HOT SAX. The next step is the application of the SAX and MSAX representations. The transformation of the raw time series data to the symbolic representations is performed in the same order as discussed earlier in this section, and thanks to the Julia programming language's ecosystem of libraries, can be easily translated into code. SAX is applied to each of the ECG leads individually, while MSAX is applied as designed to both at once. HOT SAX comprises the next step. For MSAX, the HOT SAX process is performed using the MSAX representation and distance measure. The method returns a list of distances as well as a list of indices that indicate which PAA segment has which distance. Depending on the parameter  $k$ , only the top  $k$  of these discords are returned. These results are then added to the respective PAA segments in the ECG data frame, adding both the MSAX distance of the segment as well as a binary indicator of whether or not the segment was detected as a discord. For SAX the process slightly different. Because SAX is a univariate representation, it cannot be directly applied to a bivariate ECG. Thus, SAX is applied to each lead of the ECG separately and HOT SAX is performed for each representation of each lead. Each set of results is, like MSAX, a list of indices of PAA segments and a list of their distances. Each sets of results is also added to the ECG data frame. This time the detection indicator is quaternary, it represents no detection, detection on the first lead, detection on the second lead, or detection on both leads. After both of these processes are completed, the ECG data frame is written to a CSV file for further analysis. This process can be repeated thousands of times to create data of different values for the parameters to determine optimal values and their influence.

**Statistical Analysis of Results** After completing the computations for different sets of parameters, the results need to be analyzed. While HOT SAX is not a classifier in the sense of classifying heartbeats by medical standards, it does classify them into discords and non-discords. Thus, it is a binary classifier. Binary classifiers can be evaluated using the well-known True Positive, True Negative, False Negative, and False Positive values. Table 1.1 shows their relationship. The values in Table 1.1 can be used to calculate many useful ratios that assist the evaluation of the HOT SAX algorithm. This research uses the recall value (also known as sensitivity), the accuracy, and the

Table 1.1: Contingency table showing the relationship between detected discords and actual annotated values.

Actual \ Assigned	Discord Detected	Non-Discord Detected
Is Discord	True Positive	False Negative
Is Non-Discord	False Positive	True Negative

precision. These ratios are calculated as follows: recall value is defined as

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}},$$

the accuracy as

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{False Positive} + \text{False Negative}},$$

and the precision as

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}.$$

Recall can be understood as a measure of how many of the actual discords were correctly assigned the label discord. This is the most important measure for the analysis of HOT SAX applied to ECGs because in a medical scenario, identifying as many possibly relevant sections of the ECG is more important than being 100% accurate in their identification. The second most important value is precision, which can be understood as a measure of how many of the detected discords are actually discords. While it is more important to identify as many discords as possible, a 100% recall rate could be achieved simply by assigning the label of discord to every element in the time series. Furthermore, detecting too many non-discords as discords makes it harder to analyze the actual discords that were highlighted. This of course is not useful, and thus the precision of HOT SAX needs to be incorporated into the analysis. Lastly, accuracy is not a very good measure for this particular application, as the majority of the segments in an ECG are non-discords and HOT SAX only detects a minority of the segments in an ECG. This leads to a high True Negative rate and thus a relatively high accuracy, even if HOT SAX did not actually detect any actual discords. Nonetheless, accuracy is a very common indicator of classifier performance and will thus be considered.

The analysis of the methods was performed using the data whose generation was discussed above. The analysis was performed using the R programming language. R is an established statistical and mathematical programming language with great support for statistical methods and tests. The first step in the analysis was the processing of the data generated using the Julia program. This consisted of calculating the True Positive, True Negative, False Negative, and False Positive values for each parameter combination and each method. A segment was considered a “non-discord” if its annotation consisted of an “N” or nothing “”. The former is obvious; the decision to consider no annotation (“”) a non-discord was made because for certain segments of the ECGs, no annotations were available. This can happen if, for example, the subsequence length for HOT SAX is much

smaller than one heartbeat. In that situation, one heartbeat might be represented by 5 or more sub-segments. The heartbeat annotation, given for a specific point in time, will only fall into one of the 5 segments and can thus only be counted for that one segment. The same is true for an annotation showing a discord. This method of analysis puts HOT SAX at a disadvantage because a discord located in one subsegment might influence its neighboring segments and thus lead to their detection. This detection might be an actual discord being detected, but counting it as one would incorrectly inflate the True Positive rate by assuming something about the data that it itself does not support without some inference. Thus the decision was made to accept lower True Positive values than may be accurate. **TODO** explain why? or do so later in limitations section. Any annotation that was not empty or “N” was considered a discord. This includes the medical annotation for arrhythmia but also the annotations for changes in signal quality or noise. This is done because HOT SAX is not meant to classify heartbeats by medical significance, but by how different they are from other heartbeats. A very noisy normal heartbeat will be detected the same as a arrhythmic beat. The classification of the detected discords into medically normal and abnormal heartbeats is left to more sophisticated analysis methods or human experts. The purpose of the HOT SAX methods is merely to reduce the number of ECG segments that need to be analyzed by pre-selecting the beats likely to contain useful information. After calculating the True Positive, True Negative, False Negative, and False Positive value for each parameter combination, they were collected in a data frame also containing information on the parameters that lead to them. These data frames are then saved as CSV files for further analysis. The contingency values were analyzed for the HOT SAX with MSAX method, for HOT SAX with individual SAX (only considering a single ECG lead), and for a HOT SAX with combined SAX method where the detected discords of the individual HOT SAX with SAX computations were combined. The very last step in the analysis was the calculation of the average recall, precision, and accuracy across all 48 ECGs for SAX and MSAX. This allows for a simpler comparison of the results for different parameters and enables pruning of certain parameter combinations before more sophisticated analysis begins.

## 1.2 Limitations of the Implementation

The limitations of this research are the following: HOT SAX is not a classifier based on medically relevant information, it classifies discords and not beat types. This means that its applications to diagnosing heart conditions is limited. HOT SAX can be used to pre-select specific ECG segments to look at and analyze because they exhibit discords, but it cannot, by itself, perform any type of diagnosis. The previous paragraph explains why empty annotations have to be counted as normal beats and while the author believes that this is necessary, it does negatively influence the results. The implementation of the representation only allows the use of PAA segment numbers that evenly divide the sampling frequency of the ECG database. This was done so that the whole ECG, being an even multiple of the sampling frequency itself, can be evenly divided into PAA segments. This decision prohibits certain numbers of PAA segments as there may be numbers that do not evenly divide into the sampling frequency but that do evenly divide the number of raw data points in the ECG. A further simplification step in the same vein is the restriction of subsequence values to

numbers that evenly divide the number of PAA segments  $w$ . This was also done to simplify the process and to guarantee that the whole ECG would be evenly divisible into subsequences.

**TODO** finish this

**TODO** move this to somewhere else? **TODO** mention the 1 second interval connection to the sampling frequency and why it works

## 1.3 Data Analysis

In this section, the results of the data analysis will be presented. For both the first and second datasets used in the research, the parameter selection and a summary of the results will be presented.

For dataset 2, the analysis will be completed in accordance with the methods laid out in section

**TODO** refer to methodology section here

### 1.3.1 Dataset 1

The first dataset is based on parameters that were arbitrarily chosen. This was done because the behavior of the methods was not yet known. While the choice was arbitrary, it was attempted to spread the values out in a reasonable way. For each parameter, Table 1.2 shows the method that it influences, the values that were chosen for it, and the rationale behind choosing those values. For SAX and MSAX,  $w$  the number of PAA segments in one second which is equivalent to dividing 360 data points by  $w$ . Thus,  $w$  has to be a factor of 360. These factors were chosen arbitrarily and kept small because it was not known how time intensive the computations would be. The second parameter is  $a$ , the alphabet size which determines the number of different symbols used in the SAX and MSAX discretization processes. Because the alphabet size is constrained by the size of the English alphabet, numbers were chosen arbitrarily in that range. For HOT SAX and HOT MSAX, the parameter  $k$  is number of discords they return. Giving it the value  $-1$  indicates that all available discords should be returned, regardless of how many there are. The values for  $k$  were chosen arbitrarily. This parameter does not affect the performance of the method, it just determines how many of the results are considered. Parameter  $m$  is chosen after parameter  $w$  for SAX and MSAX and represents the number of PAA segments that are grouped together to form a HOT SAX or HOT MSAX subsequence. This parameter must evenly divide  $w$ . Using these

Table 1.2: Table of the methods used for dataset 1. the parameters of each method, the rationale behind the parameter choice, and the values the parameter takes are shown.

Method	Parameter	Rationale	Values
SAX/MSAX	$w$	arbitrary factors of 360	2, 3, 4, 5, 12, 20, 30, 40, 60
	$a$	arbitrary, $2 \leq a \leq 25$	4, 5, 6, 7, 8, 9, 10, 12, 14, 17, 20
HOT SAX/MSAX	$k$	arbitrary	-1, 25, 50, 100, 150, 200, 300, 500
	$m$	arbitrary factors of 360 and of $w$	2, 3, 4, 5, 12, 20, 30, 40, 60

parameters and the programs created as part of this research, the **TODO** refer to the section that

explains the programs 48 ECGs of the MIT-BIH database were analyzed using HOT SAX and HOT MSAX. The 2,640 unique sets of parameters resulting from Table 1.2 applied to 48 ECGs creates a dataset with 126,720 files. These files were analyzed as stated in **TODO** cite methods statistics section. The mean values of the statistical measures for each set of unique parameters were calculated. Then, the threshold of recall  $\geq 95\%$  was applied to the summarized data to select the parameter combinations that yield acceptable results. Upon further analysis, it was noted that most of the parameter combinations that achieved recall  $\geq 95\%$  had  $m = w$ . To investigate this, the parameter combinations with  $m \neq w$  and recall  $\geq 95\%$  were extracted. Table 1.3 shows the results of this analysis. As Table 1.3 shows, less than 1% of the parameter sets have a recall of  $\geq 95\%$ .

Table 1.3: Results of the analysis of dataset 1. The total number of parameter sets and the number and proportion of parameter sets in dataset 1 that fulfill the conditions analysis are presented for each method.

Method	Total Sets	Sets Satisfying Analysis Conditions	
		recall $\geq 95\%$	recall $\geq 95\%$ and $m \neq w$
S-SAX	2,640	3 (0.1%)	0 (0%)
D-SAX		13 (0.5%)	0 (0%)
MSAX		23 (0.9%)	3 (0.1%)

$\geq 95\%$ , regardless of the method used. Additionally, only 3 (0.1%) parameter sets for MSAX have the desired recall and use a value for  $m$  that is different from  $w$ . These sets of results are too small to continue this analysis. Thus, a second dataset needs to be computed. The data presented in Table 1.3 does show that using subsequence lengths  $m$  for the HOT SAX and HOT MSAX methods that are not equal to the PAA segment count  $w$  is not effective. According to these findings,  $m$  will be set equal to  $w$  for the computation of dataset 2. **TODO** support this conclusion by referring to the methods section where I say that I want to look at the top 10 values, here that is hardly possible

### 1.3.2 Dataset 2

Based on the analysis of dataset 1, the parameter selection for dataset 2 is optimized. As the value of  $m$  is set equal to the value of  $w$ , the complexity of the computation is reduced dramatically. This is caused by two things. Firstly, the total number of parameter sets is decreased when factors  $m$  of  $w$  are not considered—the total number of parameter sets that have to be used it decreased. Secondly, dividing the number of PAA segments in 1 second into multiple subsegments increases the number of subsequences that HOT SAX and HOT MSAX need to work with and thus the complexity of the computation. By not doing that, the increase in complexity is avoided. As a result of this reduction in complexity, a larger set of the other parameter was considered. The parameters chosen for dataset 2 are shown in Table 1.4. For the SAX and MSAX parameters, all possible values were considered. Parameter  $w$  can be all factors of 360. For the alphabet size  $a$ , all possible values were considered. The HOT SAX and HOT MSAX parameters were chosen as follows. Parameter  $k$  was again assigned arbitrary values that provide a decent coverage for reasonable values. The value of  $-1$  is again included to signify the use of all available discords. Parameter  $m$  does not

need to be chosen for this dataset, as it is always set to the value of  $w$ . All parameters used in dataset 2 have the same meaning as in dataset 1, please refer to that section for their explanations. This table provides values that create 4,968 parameter combinations when used with this research's

Table 1.4: Table of the methods used for dataset 2. The parameters of each method, the rationale behind the parameter choice, and the values the parameter takes are shown.

Method	Parameter	Rationale	Values
SAX/MSAX	$w$	factors of 360	2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 18, 20, 24, 30, 36, 40, 45, 60, 72, 90, 120, 180, 360
	$a$	$2 \leq a \leq 25$ , length of alphabet	$\overline{2, \dots, 25}$
HOT SAX/MSAX	$k$	arbitrary	-1, 25, 50, 75, 100, 150, 175, 200, 300
	$m$	same as $w$	see $w$

programs to analyze the 48 ECGs of the MIT-BIH database. As a result, dataset 2 contains 238,464 files of ECGs analyzed with HOT SAX and HOT MSAX. Each of those combinations was applied to all 48 ECGs. As with dataset 1, these files were analyzed as stated in **TODO** cite methods statistics section. The mean values of the statistical measures for each set of unique parameters were calculated and the threshold of recall  $\geq 95\%$  applied. Table 1.5 shows the results of that analysis. As Table ?? shows, of the 4,968 total parameter sets, 99 have a recall of  $\geq 95\%$  for

Table 1.5: Results of the analysis of dataset 2. The total number of parameter sets and the number and proportion of parameter sets fulfilling the conditions that recall be  $\geq 95\%$ .

Method	Total Sets	Sets Satisfying recall $\geq 95\%$
S-SAX	4,968	99 (1.2%)
D-SAX		192 (3.9%)
MSAX		255 (5.1%)

S-SAX, 192 for D-SAX, and 255 for MSAX. These dataset are large enough for the analysis to continue. As discussed in **TODO** reference the methods statistics section, the subsets of dataset 1 for which the recall is  $\geq 95\%$  will further be sorted by the precision, in descending order. Then, the top 10 values of S-SAX, D-SAX, and MSAX will be analyzed individually.

### 1.3.3 Analysis of S-SAX

The top 10 values of S-SAX were first pruned by a threshold of recall  $\geq 95\%$  and then sorted descendingly by precision. Table ?? provides an overview of the parameters of these top 10 values, as well as their recall, accuracy, and precision. Table ?? shows that for the top 10 values by precision, the recall values are all approximately 95%, the accuracy is between 36% and 38%, and the precision is 36%. It is notable that all  $k$  values are -1. To be able to choose a best parameter combination of these 10, their interquartile ranges and outliers for the recall value will be compared

Table 1.6: Table presenting a ranking of the top 10 most precise S-SAX parameter combinations and their parameters  $k$ ,  $w$ ,  $m$ , and  $a$ . The recall and accuracy values are also shown.

Rank \ Properties	$k$	$w$	$m$	$a$	Recall (%)	Accuracy (%)	Precision (%)
1	-1	40	40	19	95.21	37.46	35.94
2	-1	24	24	24	95.19	37.57	35.93
3	-1	30	30	22	95.37	37.46	35.93
4	-1	36	36	20	95.09	37.41	35.92
5	-1	45	45	18	95.24	37.33	35.89
6	-1	24	24	25	95.74	37.32	35.88
7	-1	72	72	15	95.02	36.86	35.87
8	-1	36	36	21	95.92	37.06	35.86
9	-1	30	30	23	95.72	37.17	35.86
10	-1	120	120	13	95.13	36.51	35.85

with the help of a boxplot. This boxplot is based on the set of 48 ECGs for each ranked method. This plot can be seen in Figure 1.1. Figure 1.1 illustrates two important things. Firstly, there are only small differences between the recall values for any of the top 10 S-SAX parameter sets. Secondly, a clear pattern of 9 outliers is visible.

To select the optimal parameter based on Figure 1.1, the interquartile range (IQR) is calculated for each rank. The use of IQR versus the mean is justified here because of the skewed data and the outliers. Ranks 6, 8, and 9 have the lowest IQR, at 0.0358, 0.0354, and 0.378, respectively. For further differentiation, the number of outliers is considered. Rank 6 has 13 outliers, rank 8 11, and rank 9 has 13. Considering these factors, rank 8 is chosen as the optimal parameter set for S-SAX. This set has the smallest IQR of the top 10 and the smallest number of outliers among those with small IQR values. With a  $w$  value of 36, rank 8 represents a 10 time dimension reduction compared to the raw data.

### 1.3.4 Analysis of D-SAX

For D-SAX, the top 10 values were also first pruned by a threshold of  $\text{recall} \geq 95\%$  and then sorted descendingly by precision. Table 1.7 provides an overview of the parameters of these top 10 values, as well as their recall, accuracy, and precision. Table 1.7 shows that for the top 10 values by precision, like in the previous section. The recall values are between 95% and 97%, the accuracy between 39% and 42%, and the precision is 36%. Again, all  $k = -1$ . A boxplot comparing recall with rank is created in order to choose a best parameter combination. This plot is shown in Figure 1.2. Visually, Figure 1.2 shows ranks 6, 7, and 9 to have the smallest IQR. This is confirmed by calculating the IQR. Rank 6 has an IQR of 0.038, rank 7 has 0.03, and rank 9 has 0.04. Regarding the number of outliers, rank 7 has 6, rank 6 has 7, and rank 9 has 8. Thus rank 7 is selected as the optimal parameter set. It has both the smallest IQR and the lowest number of outliers compared to other ranks with low IQRs. The dimension reduction ratio of rank 7 is 30.

**Boxplot of Recall for 10 best S-SAX Parameter Sets**

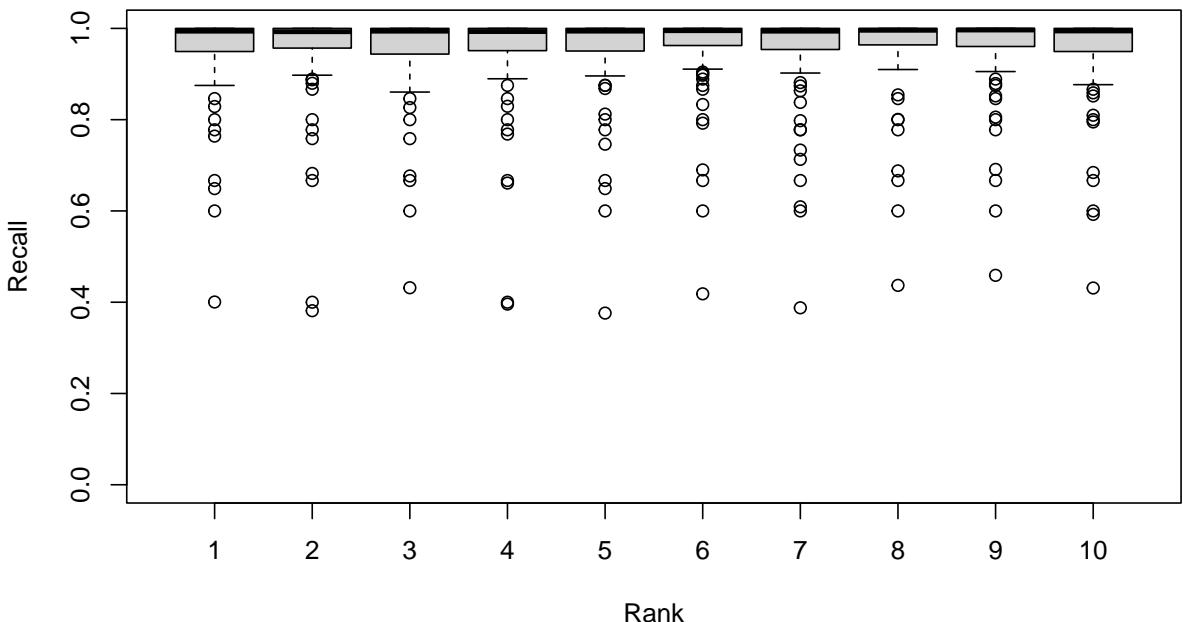


Figure 1.1: Boxplot showing the recall for the top 10 S-SAX parameter sets. The full list of parameters can be found in Table 1.6.

### 1.3.5 Analysis of MSAX

Lastly, the top 10 parameter sets of MSAX are considered. They, too, were selected based on recall  $\geq 95\%$  and sorted by precision. Table 1.8 shows the parameter set and statistical measures associated with the top 10 MSAX parameter combinations. Table 1.8 shows, for the third time, that there is little difference between the statistical results of the top 10 best parameter sets. The recall values are all 95% to 96%, the accuracy is between 38% and 41%, and the precision is 36%. A boxplot is constructed for these 10 parameter sets to explore the interquartile range and outliers of MSAX. Figure 1.1 shows this boxplot. In Figure 1.3, ranks 6, 8, and 10 visually stick out as having the lowest IQR. Calculating the IQR confirms that, yielding 0.038, 0.033, and 0.03 for ranks 6, 8, and 10 respectively. The number of outliers for rank 6 is 7, for rank 8 it is 6, and for rank 10 it is 8. Because rank 8 has the second lowest IQR and the smallest number of outliers, it is selected as the optimal parameter set for MSAX. Its  $w$  value of 12 results in a dimension reduction of 30 compared to the raw data.

### 1.3.6 Comparison of Optimal Parameters

After determining the optimal parameter sets for the three methods investigated here, they will now be compared with each other. Table 1.10 summarizes the parameter sets that are optimal for S-SAX, D-SAX, and MSAX. When it comes to the method parameters, it is desirable for  $w$  to be as small as possible to take advantage of the dimension reduction properties of SAX and MSAX.

Table 1.7: Table presenting a ranking of the top 10 most precise D-SAX parameter combinations and their parameters  $k$ ,  $w$ ,  $m$ , and  $a$ . The recall and accuracy values are also shown.

Rank \ Properties	$k$	$w$	$m$	$a$	Recall (%)	Accuracy (%)	Precision (%)
Rank							
1	-1	12	12	22	95.28	41.91	36.56
2	-1	10	10	25	95.18	41.99	36.53
3	-1	15	15	19	95.14	41.70	36.46
4	-1	12	12	23	95.18	41.00	36.34
5	-1	40	40	12	95.08	39.64	36.29
6	-1	15	15	20	96.11	40.31	36.27
7	-1	12	12	24	97.04	40.16	36.26
8	-1	36	36	13	95.80	38.93	36.20
9	-1	20	20	17	95.87	39.82	36.19
10	-1	30	30	14	96.09	39.32	36.18

Table 1.8: Table presenting a ranking of the top 10 most precise MSAX parameter combinations and their parameters  $k$ ,  $w$ ,  $m$ , and  $a$ . The recall and accuracy values are also shown.

Rank \ Properties	$k$	$w$	$m$	$a$	Recall (%)	Accuracy (%)	Precision (%)
Rank							
1	-1	6	6	24	95.37	40.68	36.24
2	-1	12	12	16	95.10	39.85	36.24
3	-1	9	9	19	95.20	39.70	36.13
4	-1	10	10	18	95.89	39.45	36.12
5	-1	8	8	21	96.01	39.53	36.12
6	-1	6	6	25	96.02	39.94	36.12
7	-1	36	36	10	95.16	38.47	36.08
8	-1	12	12	17	96.51	38.89	36.06
9	-1	30	30	11	95.49	38.26	36.03
10	-1	72	72	8	95.70	37.74	36.03

Accordingly, D-SAX and MSAX have the best  $w$  parameters, as they are smaller than  $w$  of S-SAX. Considering parameter  $k$ , they all have  $k = -1$ . The alphabet size  $a$ , while not influencing the dimension reduction, is relevant for the complexity of the method—lower values of  $a$  result in a simpler representation. Accordingly, MSAX has the best value for  $a$  because it is smaller than the competing methods. To further compare these sets of parameters, a boxplot of their recall values is created. Figure 1.4 shows this boxplot. Visually, the IQR of D-SAX look to be the smallest, S-SAX the largest. The outlier spread is also larger for S-SAX than for the other methods. The important variables regarding Figure 1.4 are summarized in Table 1.10. Table 1.10 shows that D-SAX has the best IQR, S-SAX has the highest median recall, and both D-SAX and MSAX have the lowest number of outliers. As a final test, a possible biserial correlation between the methods and the resulting recall is investigated using R’s standard `cor()` function, which calculates the Pearson correlation. The correlation coefficient of the recall depending on S-SAX or D-SAX is 0.06, for

**Boxplot of Recall for 10 best D-SAX Parameter Sets**

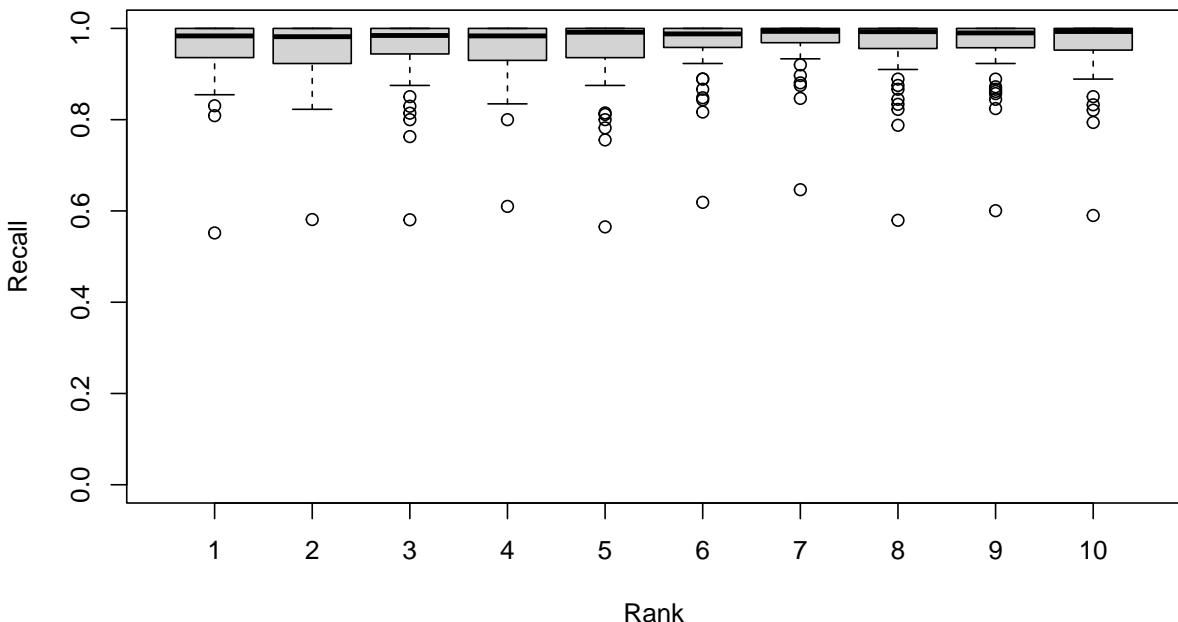
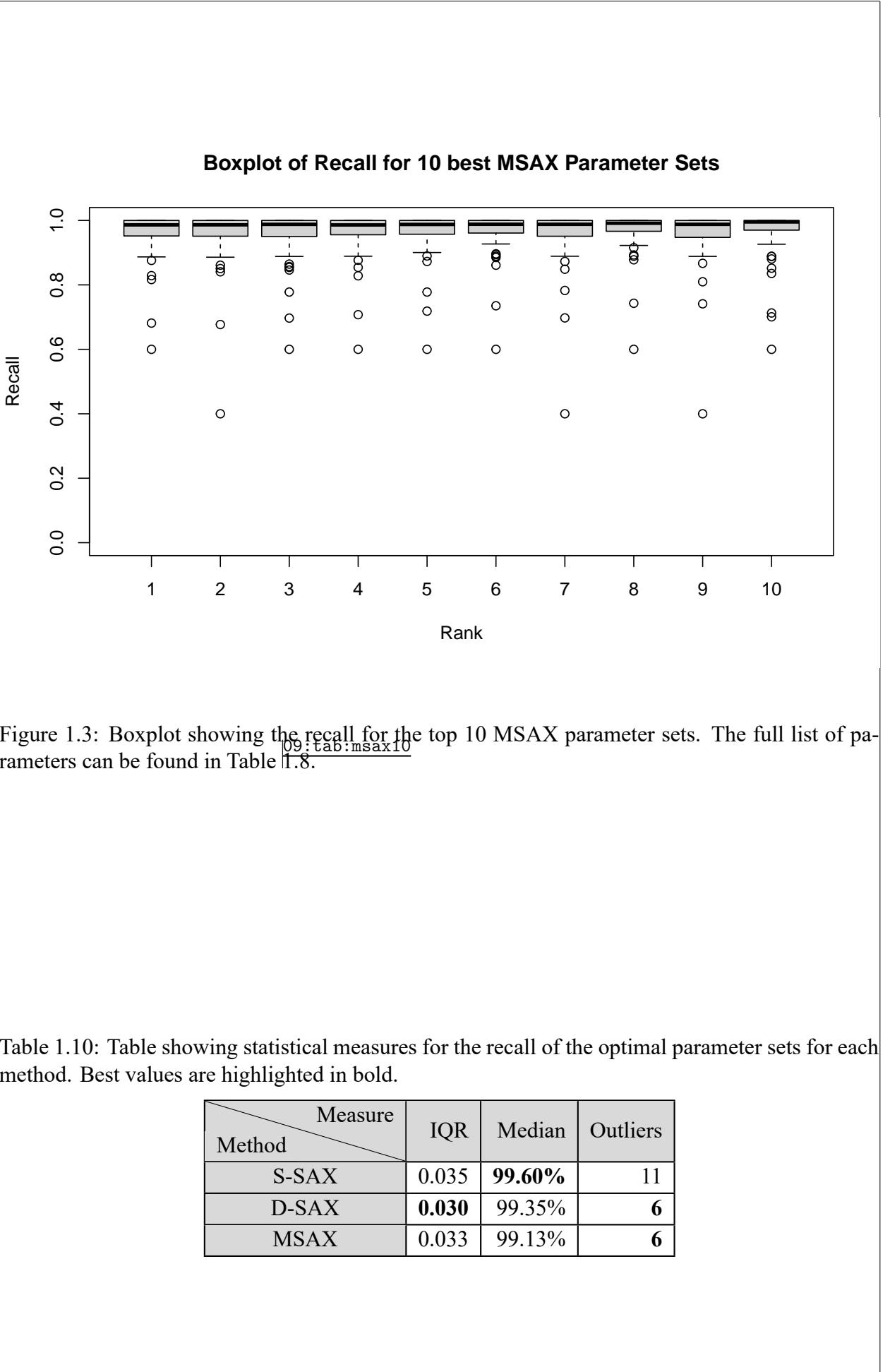


Figure 1.2: Boxplot showing the recall for the top 10 D-SAX parameter sets. The full list of parameters can be found in Table 1.7.

Table 1.9: Table showing the optimal sets of parameters for S-SAX, D-SAX, and MSAX. Parameters  $w$  and  $m$  are combined because they are equal to each other. Best parameters are highlighted in bold.

Method \ Parameter	$k$	$w, m$	$a$
S-SAX	-1	36	21
D-SAX	-1	<b>12</b>	24
MSAX	-1	<b>12</b>	<b>17</b>

the recall depending on S-SAX or MSAX it is 0.033, and for the recall depending on D-SAX or MSAX it is -0.04. All three correlation coefficients are very close to zero, which indicates that there is no correlation between the methods used and the resulting recall value.



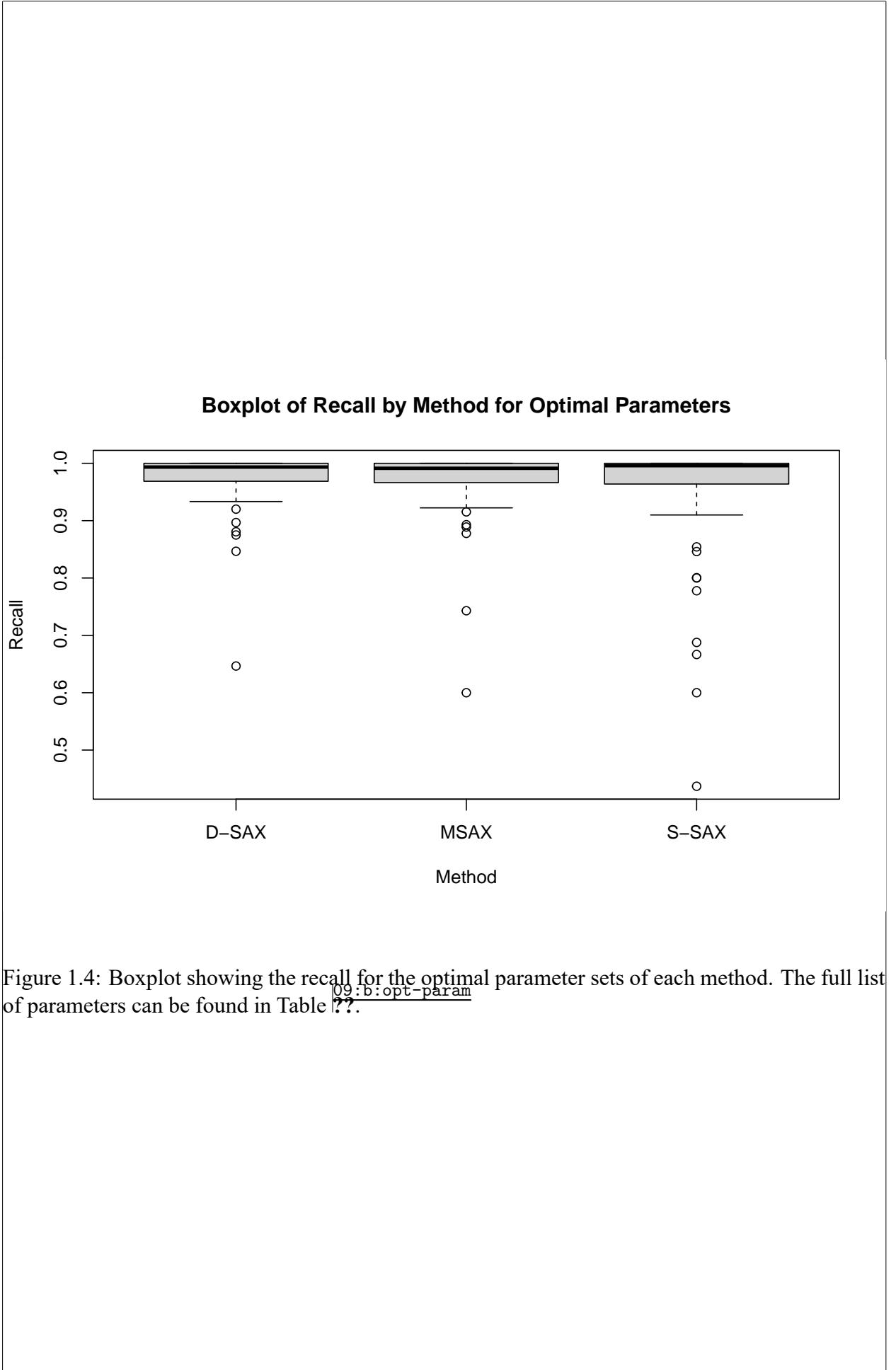


Figure 1.4: Boxplot showing the recall for the optimal parameter sets of each method. The full list of parameters can be found in Table [09:b:opt-param](#).

09:fi