UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola d'Enginyeria de Barcelona Est

MASTER'S THESIS

Master's degree in Interdisciplinary & Innovative Engineering

ANALYSIS OF CLINICAL DIAGNOSTICS IN HUMAN GENETICS WITH SEMANTIC SIMILARITY SEARCHES IN ONTOLOGIES

Report and annexes

Author: Rafael León Carrasco
Director: Alexandre Perera Lluna
Co-director: Jon Garrido de Aguirre

Date: June 2022

Abstract

This thesis is focused on the analysis of the results of the paper *KÖHLER, Sebastian, et al. Clinical diagnostics in human genetics with semantic similarity searches in ontologies* (1) and the improvement of them.

The method studied uses symptoms from the *"Human Phenotype Ontology" 2022 (HPO)* (2) and tries to predict the disease using another database from the same website (3). This comparison is performed using semantic similarity methods that will be explained in the next chapters.

While trying to reproduce the results, many difficulties were found to reach that goal. After many auditions of the code that run the simulation, the performance was not close to that shown in the paper. However, an implementation of this method is already available online by the own authors of the article. This is called the *Phenomizer*. After generating hundreds of simulated patients following the rules specified in the paper, and using this data in the *Phenomizer*, one conclusion can be extracted from it, the results are not consistent with the data specified in the article. When imprecision is added to the symptoms (randomly replacing a symptom by one of their ancestors), performance is much worse than stated. On the other hand, adding noise (symptoms unrelated to the disease) has little effect on the output and the performance is coherent with the paper.

To try to improve the performance, some other semantic similarity methods were tested using a specialized library called *Semantic Measures Library and Toolkit* (3). No significant improvements were detected in this investigation, the method used in the article was the best in terms of results quality.

The last important point that is clashing with the article is the fact that they use a p-value calculated from the semantic similarity score to obtain the rankings of more probable diseases for the given set of symptoms. According to the paper, the p-value method is better than using simply the similarity scores. However, in all the tests done in this thesis similarity scores performed better than p-values. This means that no big calculations, expensive in time, and a big hard drive capacity are required.

In conclusion, the results of the real *Phenomizer* are worse than stated; no better semantic similarity method than the existent one was found, and disease rankings by similarity scores performed better than by p-value, contrary to the paper assertions.

Acknowledgments

Big thanks to Jon Garrido for giving me the idea for this thesis and for helping me understand all the details of the paper on which this project is based before starting working on it. Moreover, thanks for providing data and the invaluable help in applying the Hochberg and Benjamini correction in the data.

Also, I'm very thankful to Alexandre Perera for presenting the idea of this thesis to me and for making time in his schedule to give me some insight into this work.

Index

ABSTRACT I

ACKNOWLEDGMENTS II

INTRODUCTION 5

1.2. Main objectives 5

1.3. Scope 5

CLINICAL DIAGNOSTICS IN HUMAN GENETICS WITH SEMANTIC SIMILARITY SEARCHES IN ONTOLOGIES 7

1.4. Ontologies 7

1.5. Similarity score 8

1.6. P-value comparison 10

1.7. Simulated patients 11

SYSTEM IMPLEMENTATION 13

1.8. Databases used 13

1.9. Score system 14

1.9.1. SML initialization with examples 14

1.9.2. Information content implementation 16

1.9.3. Similarity score calculation 18

1.10. P-values calculation 19

1.11. Simulation of the patients 26

1.11.1. Dysmorphology syndromes data 26

1.11.2. Patients' generation 27

1.11.3. Noise and imprecision 28

1.11.4. Benjamini and Hochberg correction 30

RESULTS 33

1.12. Results of the system implemented in this thesis and comparisons 34

1.13. Evaluation of Phenomizer using web scraping 42

1.14. P-value reliability 48

1.15. More similarity score methods 53

1.15.1. Direct Groupwise measures 54

1.15.2. IC-based measures 57

CONCLUSIONS 63

BIBLIOGRAFIA 65

ANNEX 67

Scores comparisons between 100000 and 5000 samples 67

1.16. Code Index 73

Limited imprecision method 74

Limit → 3 74

Performance comparison between old and latest databases 78

Phenomizer vs local simulation 79

Latest databases comparison 79

Older databases comparison 81

P-value Truth Tables 84

Similarity score methods 91

IC-based measures 105

Introduction

The most important part of the job of a physician is making the correct diagnosis because identifying some diseases that can be caused by a big number of different genetic disorders is complicated. Especially when many of these diseases share many common symptoms. Moreover, the same disease can express different symptoms in different patients, plus the added difficulty that some of the identified symptoms can be unrelated to the disease. A correct diagnosis can help save time and start the appropriate treatment soon.

To ease those difficulties a method for clinical diagnostics was developed in 2009 by a team led by Sebastian Köhler, using the semantic structure of the Human Phenotype Ontology (HPO) (1). According to the paper, the method works as follows:

*"A p-value is assigned to the score obtained by searching on **n** terms, corresponding to the probability of obtaining a given similarity score or better by choosing the same number of query terms at random"* (Köhler et al. 2009)

A lower p-value is a good differential diagnostic for the set of symptoms selected. On the other hand, a higher p-value means that the symptoms are not specific enough to allow a good diagnosis, or that the disease is not described in the database used for the calculations.

1.1. Main objectives

- Reproduce the results of (Köhler et al. 2009).
- Try to improve the results using many different semantic similarity methods apart from the ones used in the paper.

1.2. Scope

Some basic concepts of (Köhler et al. 2009) will be described to give a better insight into the procedures followed in this work.

To reproduce the results of the paper, some code had to be written to simulate the characteristics and rules defined in the paper, because there was no open-source code available for this goal.

Once the results are correctly reproduced, many new semantic similarity measures will be tested and the conclusions presented.

Clinical diagnostics in human genetics with semantic similarity searches in ontologies

This chapter will focus on the key points of (Köhler et al. 2009) that will be used in this thesis to create a good simulation of the process and replicate the results for scrutiny. With this, some improvements will be proposed. Also, some basic terms and definitions are provided to better understand this work.

1.3. Ontologies

The paper's method is built onto an acyclic ontology system which represents symptoms that can be referenced by the disease database called Human Phenotype Ontology (HPO) (1). Each concept of the ontology used is a symptom, this symptom can be preceded by one or more general concepts, or it can stem from more specific terms like in Fig. 1. Another point is that in the paper (Köhler et al. 2009) and this thesis only the symptoms that are children of *Phenotypic abnormality* are used, while the other unused branches are: *Blood group, Clinical modifier, Frequency, Mode of inheritance, and Past medical history*; which are not relevant for the scope of this work.

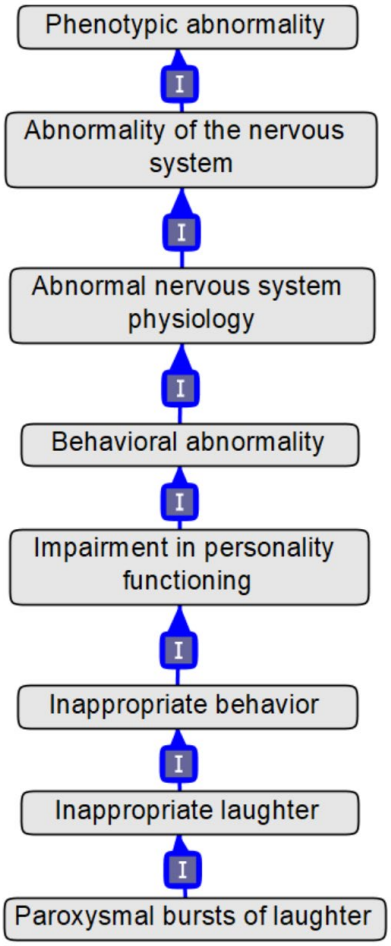


Figure 1. This is an example of one part of the ontology used. Symptoms are represented hierarchically. General terms are at the top of the graph while the specific terms are at the bottom. One symptom can have more than one parent or children.

The terms of the HPO are used for the annotations of diseases described in another database. Each disease is related to a set of symptoms that are present in the HPO. The databases used in Köhler et al. 2009 are old versions that contain nearly 9000 symptoms for the HPO and 4813 diseases for annotations. In the following chapters, more up-to-date versions are used and some conclusions about performance are extracted in comparison with the 2009 databases.

1.4. Similarity score

The method described in Köhler et al. 2009 uses a query, a set of symptoms for which you want to know which disease corresponds to it. For this goal, a score must be returned for every disease in the database and the higher one should be the correct diagnosis.

First, the specificity is taken into account to obtain the information content of the symptom or *IC*. The less common a symptom in the annotations database, the higher the score. For example, if a symptom

like a *paroxysmal burst of laughter* is only annotated in 4 different diseases, for 4813 diseases the result would be -ln(4/4813) = 7.09. On the other hand, a more common symptom like *abnormality of the nervous system* is present in 51 diseases, thus the IC score would be -ln(51/4813) = 4.54.

To assess the similarity between the two terms, the most informative common ancestor (MICA) is used. MICA is an algorithm that finds the ancestor that is common to both terms and has a maximum IC value.

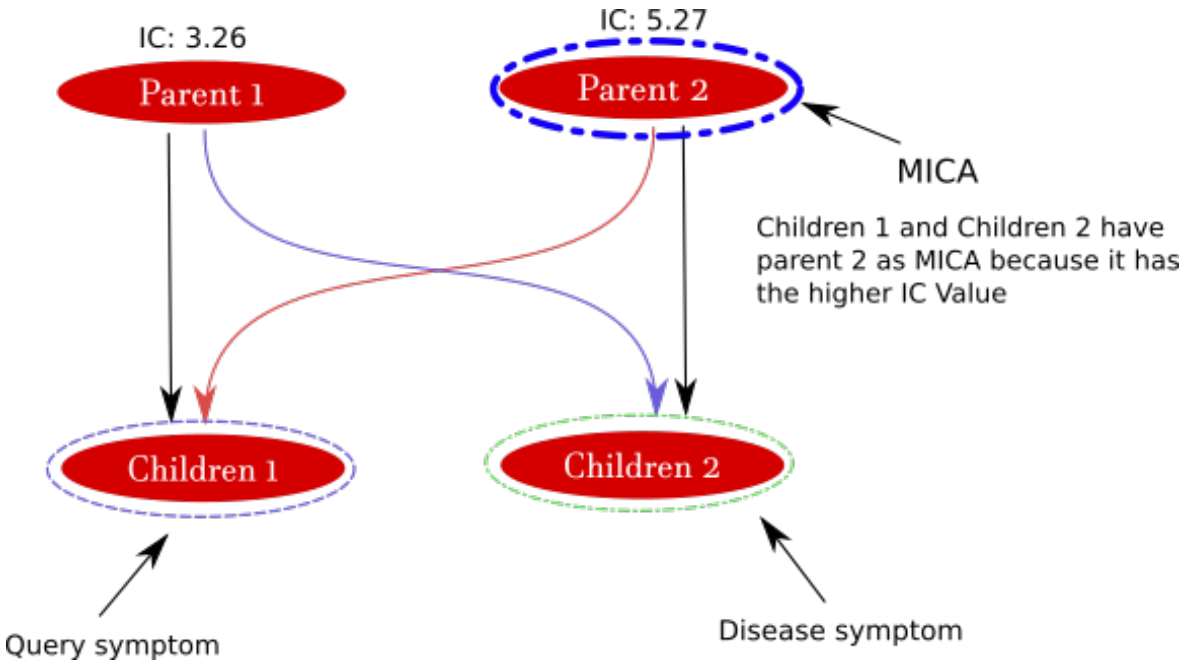


Figure 2. Example of how MICA works for a pair of symptoms being compared. Children 1 is a symptom that pertains to the query set, and Children 2 to the disease set.

With the measures established above, a method to obtain the similarity between query terms and diseases is developed by Köhler et al. 2009.

$$sim(Q \rightarrow D) = avg\left[\sum_{t_1 \epsilon\, Q} \max_{t_2 \epsilon\, D}(MICA(t_1, t_2))\right] \quad \textbf{(Equation 1)}$$

Equation 1 describes how the score is obtained after a query is submitted. Each symptom of the query is compared with all the symptoms of the selected disease using the MICA algorithm. The higher score obtained is selected and saved. Then, the next term of the query is processed in the same way until no more query terms are remaining. Finally, the saved scores are averaged and the final result for the comparison between the query and disease is returned. The query must be compared this way with all the diseases in the database to know which one is more similar to the query.

There is another equation in Köhler et al. 2009 called symmetric similarity score, but this won't be used in the paper for calculating results and therefore not in this thesis either.

1.5. P-value comparison

Is difficult to say when a score is a good match or not, because it depends on the number of symptoms or specificity of the disease that the query is being compared to. For example, a 2.3 can be a good result for Opitz syndrome, because normally the scores for this disease go around 1.7. But 2.3 can be a bad result for Noonan syndrome, with average values around 2.7.

This problem is why Köhler et al. 2009 developed a statistical method to deal with this. They estimated a p-value for each search that indicates the probability of obtaining the same or higher value compared with a set of random query terms of the same size. To obtain the different p values, they need to use a Monte Carlo random sampling approach, followed by a Benjamini and Hochberg correction. A complete calculation of all the possible scores for each disease is not possible because the number of combinations grows exponentially with the number of query terms. They needed to calculate each disease's p-values from 100000 random searches. This simulation is repeated for query sizes from 1 to 10, because for the same disease different sizes will return different average scores. If the query size is 11 or greater, the p values are calculated using the calculated distribution for 10 terms.

For example, for a query size of 3, 100000 comparisons are performed using random query sets of 3 symptoms. From this, 100000 similarity scores are obtained; then, is just a matter of ordering the 100000 scores in ascendant order, assigning p-values to each score going from 1 for the lowest to 0 for the higher value; is simply a proportion. Finally, use a Benjamini and Hochberg correction.

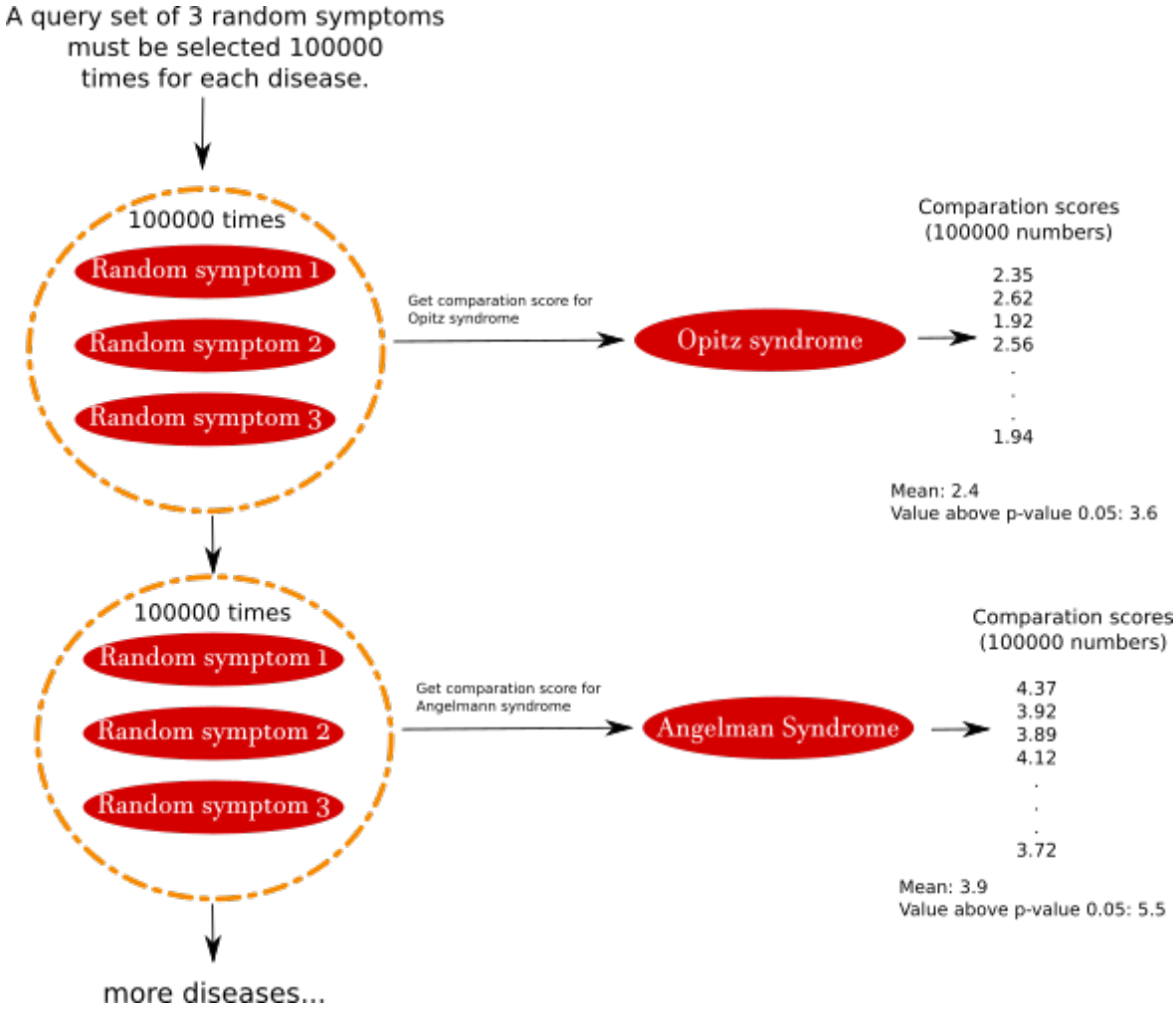


Figure 3. Example of p-value calculation for query size of 3. Similarity scores of Opitz syndrome are lower than Angelmann's, thus relevant p-values for Opitz are from 3.6 onwards, while for Angelmann are higher, from 5.5.

1.6. Simulated patients

To evaluate the performance of the algorithm developed, Köhler et al. 2009 used a set of generated patients using 44 identified complex dysmorphology syndromes that contained frequency data for their symptoms (5). They assumed no interdependency between symptoms because not enough data is available on this matter. Using the frequency information, 100 random patients are generated for each one of the 44 sets. Gender-specific features are also considered.

External symptoms not related to the real disease are added to simulate unrelated clinical problems, this is called noise. The number of noise symptoms is half the quantity of the disease-related terms.

Finally, to simulate the clinical descriptions discrepancies about the symptoms made by the physician, another variable called imprecision is added. This may be caused by the lack of knowledge about the

correct terminology used in the database, or because the symptom cannot be described easily without more advanced clinical investigations. For this, every feature of the patient is randomly replaced by one of its ancestors except the root of the ontology (Phenotypic abnormality).

System implementation

The system described in Köhler et al. 2009 is implemented in this chapter. The HPO, annotation database, the score equation (Eq. 1), and the p-value calculation will be explained thoroughly in the following pages.

1.7. Databases used

In Köhler et al. 2009, two databases are used: one for the Human Phenotype Ontology (HPO) and another for the annotation of all diseases. The HPO database is from May 6, 2009: version 1.58. In this thesis, version 1.59 (June 12, 2009) is used for proximity because 1.58 is not available (6). Also, the most recent version will be used and compared with 1.59 for completion (Chapter 4.1).

Regarding the annotation database, no reference is provided in the paper, only that contains 4813 diseases and that was updated by that date (around 2009). Unfortunately, no history of old versions of this database is available right now. The old repositories that maintained these databases are down in the meantime (7). In this case, the most recent database will be used (3). In the next chapter, 4.2 Evaluation of Phenomizer using web scraping, a method to obtain a closer version in time to the original is described and the results are compared.

Due to compatibility with the library used described in subchapter 3.2 *Score system*, the annotations database is converted from *hpoa* to *tsv* format; and only entries from the OMIM database with aspect *P* (Phenotype) are included. For more information about this format consult: (10).

For example, those are some entries in *hpoa* format for disease with id 609115 and name *Limb-girdle muscular dystrophy, type 1G* from OMIM database with their corresponding symptoms:

```
OMIM:609115 "Limb-girdle muscular dystrophy, type 1G"  HP:0001265 OMIM:609115 IEA P
OMIM:609115 "Limb-girdle muscular dystrophy, type 1G"  HP:0003236 OMIM:609115 TAS P
OMIM:609115 "Limb-girdle muscular dystrophy, type 1G"  HP:0008948 OMIM:609115 IEA P
OMIM:609115 "Limb-girdle muscular dystrophy, type 1G"  HP:0003198 OMIM:609115 TAS P
OMIM:609115 "Limb-girdle muscular dystrophy, type 1G"  HP:0003581 OMIM:609115 IEA C
OMIM:609115 "Limb-girdle muscular dystrophy, type 1G"  HP:0000518 OMIM:609115 TAS P
OMIM:609115 "Limb-girdle muscular dystrophy, type 1G"  HP:0003829 OMIM:609115 TAS M
OMIM:609115 "Limb-girdle muscular dystrophy, type 1G"  HP:0003749 OMIM:609115 IEA P
OMIM:609115 "Limb-girdle muscular dystrophy, type 1G"  HP:0003805 OMIM:609115 TAS P
OMIM:609115 "Limb-girdle muscular dystrophy, type 1G"  HP:0008116 OMIM:609115 IEA P
```

OMIM:609115 means that this disease corresponds to the OMIM database, contains 10 symptoms that are identified as *HP:XXXXXXX,* and only entries with the modifier aspect equal to P are considered because it means that they are located in the Phenotypic abnormality subontology.

Then, the *hpoa* format is translated to *tsv* to become readable by the library used in this thesis to manage the ontologies. Reference to code in annex section *Code Index.*

```
OMIM:609115
HP:0003236;HP:0000518;HP:0008948;HP:0008956;HP:0001265;HP:0003547;HP:0006203;HP:0
003198;HP:0006785;HP:0003749;HP:0008116;HP:0003805;
```

1.8. Score system

To build a reliable score system, a good method to process and interpret the ontology structure is required. For this, a library called *Semantic Measures Library (SML)* (9) will be used. The original link of the library is currently offline, so it was downloaded from here: (8).

1.8.1. SML initialization with examples

To clarify a bit the usage of this library in this thesis, some examples are provided.

```
"""
First, two databases are required to make this system work:

-The Human Phenotype Ontology (hp.obo)

-And the disease annotations database (Phenotype.tsv)
as described in the databases chapter
"""
String hpoOBO = "D:\\Users\\Rafaa\\HPO_big_files\\hp.obo";
String annot = "D:\\Users\\Rafaa\\HPO_big_files\\Phenotype.tsv";

"""
The format of the URIs used in the library must be
defined beforehand, in this case, 'http://hp/'
"""
URIFactory factory = URIFactoryMemory.getSingleton();
URI graph_uri = factory.getURI("http://hp/");

"""
The symptoms from HPO start with 'HP' as a prefix, like HP:0003281. This must be
specified.
"""
factory.loadNamespacePrefix("HP", graph_uri.toString());
```

```
G graph = new GraphMemory(graph_uri);

"""
The data from the two databases are loaded into the library.
"""
GDataConf hpoConf = new GDataConf(GFormat.OBO, hpoOBO);
GDataConf annotConf = new GDataConf(GFormat.TSV_ANNOT, annot);
annotConf.addParameter("header", "false");

GraphLoaderGeneric.populate(hpoConf, graph);
GraphLoaderGeneric.populate(annotConf, graph);
```

Code 3.1. SML initialization

SML needs some extra configuration to work properly. The two databases, annotations and HPO, must be linked by them using a virtual root.

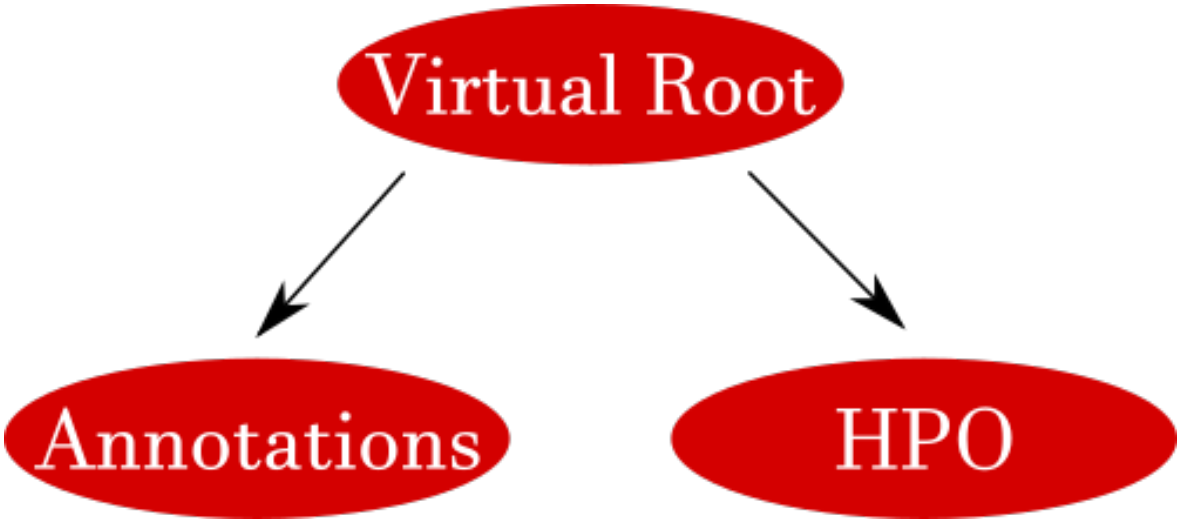


Figure 4. Virtual root function

```
"""
Create the virtual root vertex
"""
URI virtualRoot = factory.getURI("http://hp/virtualRoot");
graph.addV(virtualRoot);
"""
Root the graphs using the virtual root as the root
"""
GAction rooting = new GAction(GActionType.REROOTING);
rooting.addParameter("root_uri", virtualRoot.stringValue());
        GraphActionExecutor.applyAction(factory, rooting, graph);
```

Code 3.2 Virtual root implementation

1.8.2. Information content implementation

Is necessary to calculate the information content (IC) of every symptom to get the final similarity score from the set of terms of the query with every disease. As previously stated, the less common a symptom in the annotation database, the higher the score.

To implement this behavior, just one line is needed in SML:

```
ICconf icConfRes = new IC_Conf_Corpus(SMConstants.FLAG_IC_ANNOT_RESNIK_1995);
```

Code 3.3 IC configuration

The line described in Code 3.3 triggers the calculation of the IC. Resuming from the previous example, if a symptom is annotated in 4 different diseases, and the total number of diseases is 4813, the result is -ln(4/4813) = 7.09. If the symptom has descendants, the number of annotations of them are included also.

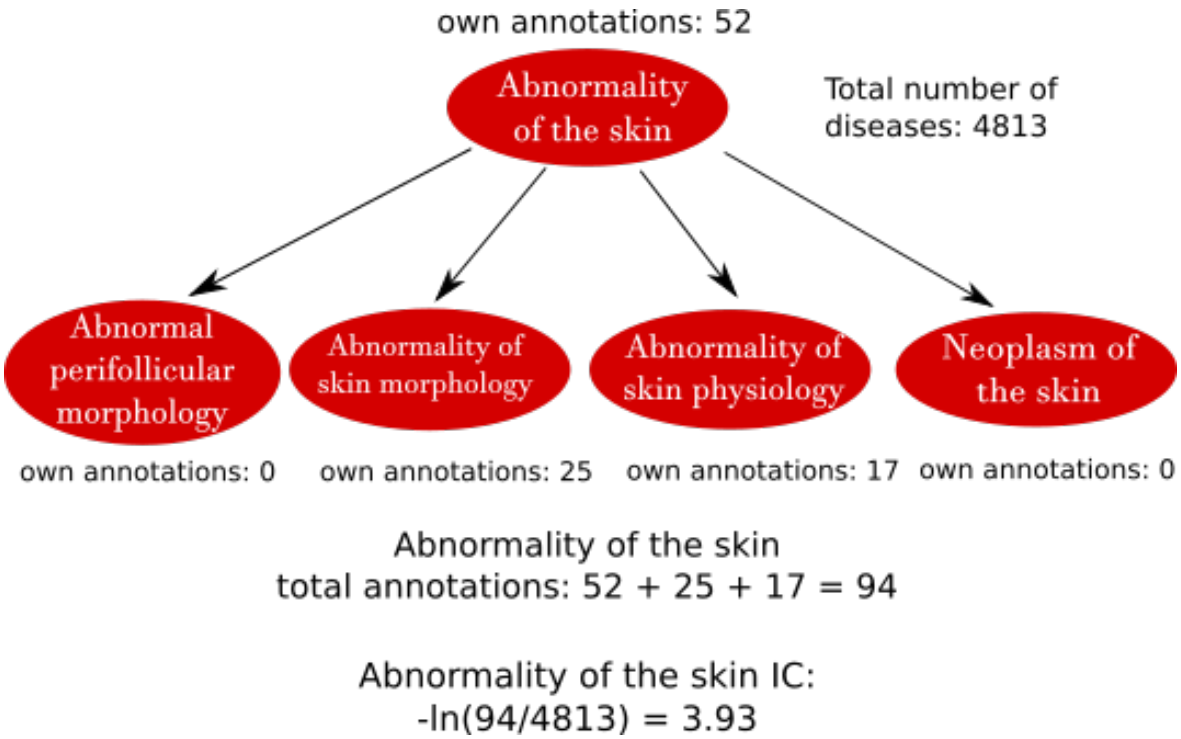


Figure 5. Example of how the number of annotations in parents is obtained. The parent inherits the number of annotations of its descendants plus their number of coincidences

However, when the number of annotated diseases is 0 for a symptom, calculating ln(0) throws an error. To avoid this, 1 is added to the number of annotated diseases for every terminal symptom (i.e. without

descendants). This implies that the number of diseases must be increased from 4813, like in the example, to 4813 plus the number of symptoms without descendants.

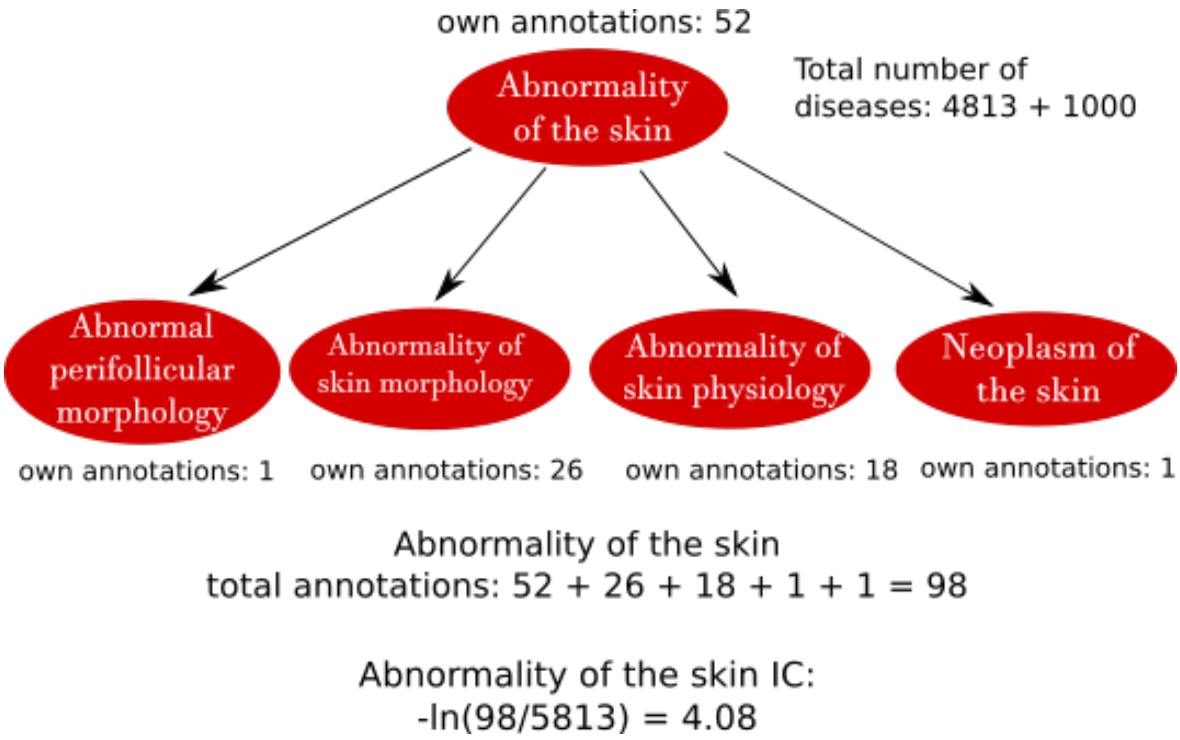


Figure 6. Example of how adding one to the terminal symptoms number is implemented. The total number of diseases is increased by 1000. This is a randomly chosen number supposed to be the total quantity of symptoms without children.

In figure 6, after adding one to the terminal symptoms, the IC value is slightly different to figure 5. However, this is not relevant, because the IC is still proportional to the number of annotated diseases.

The code showing the implementation of the IC calculation is the following:

```
     * @param nbOccurences the number of occurrences for each disease. For each
     * disease the number of occurrences must be greater than 0.
     * @return the IC of all URIs specified in the given map.
     * @throws SLIB_Ex_Critic
     */
    public Map<URI, Double> compute(SM_Engine manager, Map<URI, Integer>
nbOccurences) throws SLIB_Ex_Critic {

        Map<URI, Double> results = new HashMap<URI, Double>();

        // This gets the total number of diseases plus the terminal symptoms
        long max = Collections.max(nbOccurrence.values());

        //Calculate the IC
        double prob_occ;
        double curIc;
        for (URI v : nbOccurrence.keySet()) {
```

```
            prob_occ = (double) nbOccurrence.get(v)/ max;
            curIc = -Math.log(pc);
            results.put(v, curIc);
        }

        return results;
    }
```

Code 3.4 IC calculation

1.8.3. Similarity score calculation

The next step is the calculation of the similarity score. The implementation is pretty much straightforward, as seen in the next example.

```
Set<URI> query = ... //(Random set of symptoms to identify which disease is more
similar to it

//The IC configuration is set, as well as the HPO simulation score calculation
SMconf smConfGroupwiseHPO = new SMconf("HPO",
SMConstants.FLAG_SIM_GROUPWISE_DAG_HPO);
smConfGroupwiseHPO.setICconf(icConfRes);

for (URI disease : engine.getInstances()) {

   //This returns the symptoms of the current disease
   Set<URI> disease_annotations = iAccessor.getDirectClass(disease);

   //The score is calculated simply by using the function 'compare',
   //the configuration, query, and disease to be compared
   sim = engine.compare(smConfGroupwise, query, disease_annotations);
   System.out.println("Similarity score is: " + sim);

}
```

Code 3.5 Similarity scores calculation example

The method *compare* in code 3.5 implements Equation 1 as shown in code 3.6. Comments have been added for clarity.

$$sim(Q \rightarrow D) = avg\left[\sum_{t_1 \epsilon\, Q} \max_{t_2 \epsilon\, D}(MICA(t_1, t_2))\right] \quad \textbf{(Equation 1)}$$

```
    @Override
```

```
    public double compare(Set<URI> query, Set<URI> disease, SM_Engine c, SMconf
conf) throws SLIB_Exception {


        double total_ic_MICA = 0;

        //Compares all query items with all disease items
        for(URI a : query) {
            double max_ic_MICA = 0;
            for(URI b : disease) {
                //Gets the MICA of query symptom 'a' and disease symptom 'b'
                double ic_MICA = c.getIC_MICA(conf.getICconf(), a, b);
                if(ic_MICA > max_ic_MICA) {
                    //Keeps the higher IC value obtained from the query item 'a'
compared
                    //to all disease symptoms
                    max_ic_MICA = ic_MICA;
                }
            }
            //The maximum IC of the MICA is accumulated here to be averaged
afterward
            total_ic_MICA += max_ic_MICA;
        }

        //Average over the query size
        return total_ic_MICA / query.size();
    }
```

Code 3.6 Equation 1 implementation

These are the basics for the computation of similarity scores. The next step to replicate the paper by Köhler et al. 2009 is to develop the p-values calculation system.

1.9. P-values calculation

The paper establishes that for the calculation of the necessary p-values, 100000 random simulations for every disease are required. This procedure must be repeated for 1 to 10 query items. All the possible scores are stored on the disk (rounded to 4 decimal places) and the associated p-values. Considering that the number of diseases is 8209 if the scores and p-values are stored in *float32* format, the disk capacity in bytes used would be: 32/8 (float32 size in bytes) · 2 (p-values and scores) · 100000 (samples per disease) · 8209 (number of diseases) · 10 (query size) = 65,672,000,000 bytes. That means that calculating this database fills around **61.1 GB** of disk memory.

The computer used for these calculations has a Ryzen 2700 CPU. Extrapolating the time required for calculating the p-values for query size 1 which elapsed 20.5 min, and considering that getting the

similarity scores from a query set of 10 items is 10 times slower than 1 item, obtaining the entire p-values database would take around **18.8 h**.

This is only for one p-value database, if new ones are required with other HPO databases, or new similarity scores methods, the disk capacity and time would be unmanageable for the currently available resources.

To save space and time in the multiple p-values databases that will be calculated, an alternative method is used. First, instead of saving one p-value per score, as stated in the paper, only some that corresponds to some predefined values are stored. In other words, a pattern like the next one will be used:

p-values pattern → **(1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.09, 0.08, 0.07, 0.06, 0.05, 0.04, 0.03, 0.02, 0.01, 0.005, 0.001, 0)**

In Köhler et al. 2009, p-values are calculated like a proportion. The lower score is assigned with a p-value of 1, while the higher with a 0. The median would correspond to a p-value of 0.5. To calculate this, the next code was used:

```
List<Float> ordered_scores = /*list of similarity scores in ascending order*/

//The p-valurs pattern to use
float[] p_values_pattern = new float[] {1f, 0.9f, 0.8f, 0.7f, 0.6f, 0.5f, 0.4f,
0.3f, 0.2f, 0.1f, 0.09f, 0.08f, 0.07f, 0.06f, 0.05f, 0.04f, 0.03f, 0.02f, 0.01f,
0.005f, 0.001f, 0f};

//number of random simulations to perform
int num_of_searches = 100000;

//The p-values pattern will be stored here
float[] main_p_values = new float[p_values_pattern.length];

for(int j = 0; j < p_values_pattern.length; j++) {
    /*
    To get the score corresponding, for example, to the p-value of 0.8, the
formula below would be
    applied this way: (100000 - 1) · (1 - 0.8) = 19999. This means that
    the p-value of 0.8 corresponds to the score at position 19999, which makes
sense
    because the score list is in ascending order
    */
    int select = (int) ((num_of_searches - 1) * (1 - p_values_pattern[j]));
    main_p_values[j] = ordered_scores.get(select);
}
```

Code 3.7 p-values pattern implementation

Code 3.7 must be repeated in each disease, and for every query size from 1 to 10.

Similarity scores corresponding to the p-values pattern

p-values pattern	*Similarity scores*
1	*0.01*
0.9	*0.01*
0.8	*0.01*
0.7	*1.36*
0.6	*1.62*
0.5	*1.71*
0.4	*1.79*
0.3	*2.03*
0.2	*2.30*
0.1	*2.73*
0.09	*2.83*
0.08	*2.84*
0.07	*2.97*
0.06	*2.97*
0.05	*3.02*
0.04	*3.19*
0.03	*3.30*
0.02	*3.47*
0.01	*4.06*

0.005	*4.91*
0.001	*5.83*
0	*6.02*

Table 1. p-values pattern correspondence with similarity scores of Cerebrooculofacioskeletal syndrome 3. To visualize better how this system works: P-value 0 corresponds to the higher value obtained in the simulation, while P-value 1 to the lower one. P-value 0.5 would correspond to the score that falls just in the middle or the median (position 50000).

With the kind of data in Table 1, p-values can be obtained for every score when requested using a simple interpolation (11).

$$y = y_0 + (x - x_0)\frac{y_1 - y_0}{x_1 - x_0} \quad \textbf{(Equation 2)}$$

Where y_0 = the lower p-value; y_1 = upper p-value; x_0 = upper similarity score; x_1 = lower similarity score; x = score to be interpolated with; y = result (p-value)

For example, using Table 1 and a score of 5.12, the p-value would be:

$$0.001 + (5.12 - 5.83)\frac{0.005 - 0.001}{4.91 - 5.83} = 0.004086 \quad \textbf{(Equation 3)}$$

Equation 3 gives a p-value of 0.004086 for a score of 5.12.

Thanks to the interpolation, a lot of space in the hard disk can be saved. Now, it will fill 32/8 (float32 size in bytes) · 2 (p-values and scores) · 22 (samples per disease using the pattern) · 8209 (number of diseases) · 10 (query size) = 14,447,840 bytes or **13.77 MB**. Moreover, the data loss in the p-values is not very significant, because to assess the performance of this system the lower precision won't change the results very much, as seen in chapter 4.3 P-value reliability.

Another problem is the computing time. Even saving a lot of space using interpolation, 100000 samples must be calculated in each cycle taking a lot of time as previously stated. Because of this, the number of random queries has been reduced from 100000 to 5000.

To assess the performance impact an example is provided comparing the scores with 5000 and 100000 samples of Larsen-Like syndrome (26 symptoms).

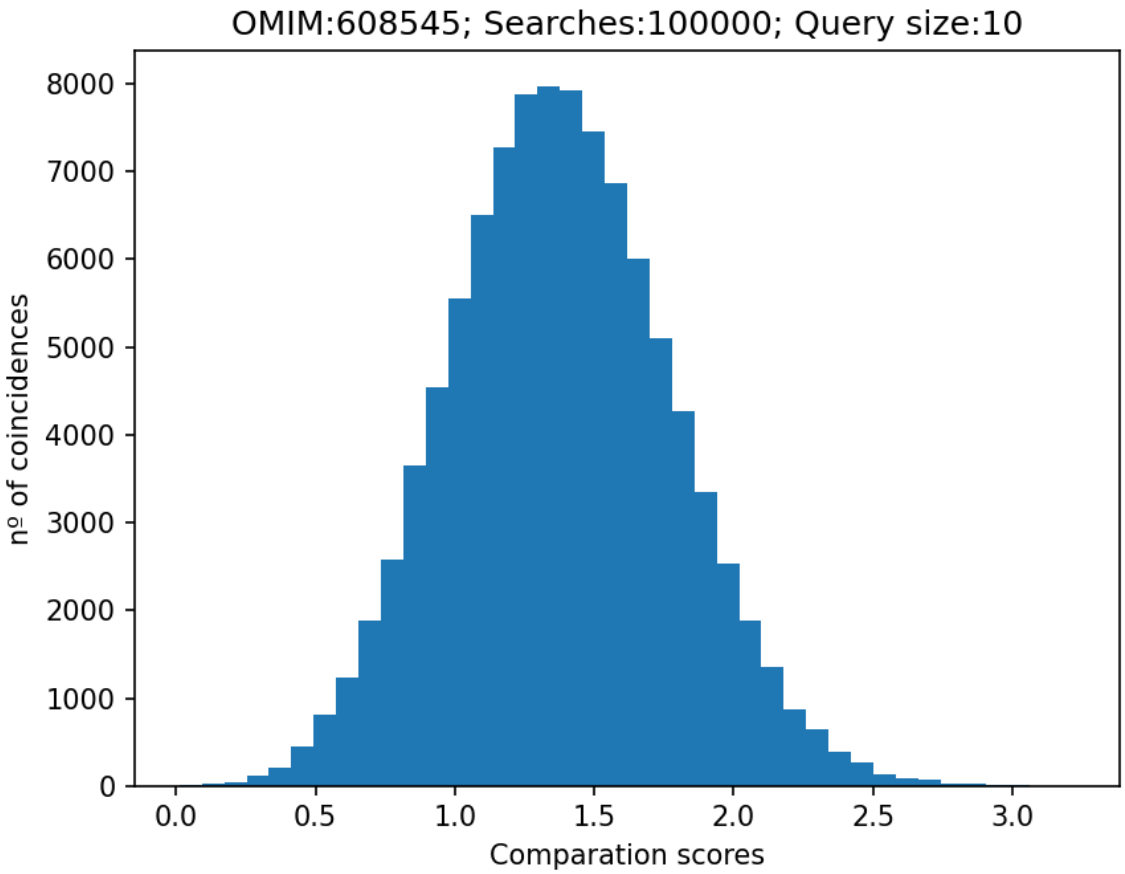


Figure 7. Histogram of the similarity scores for disease OMIM:608545 (Larsen-Like syndrome) with 100000 searches and query size of 10

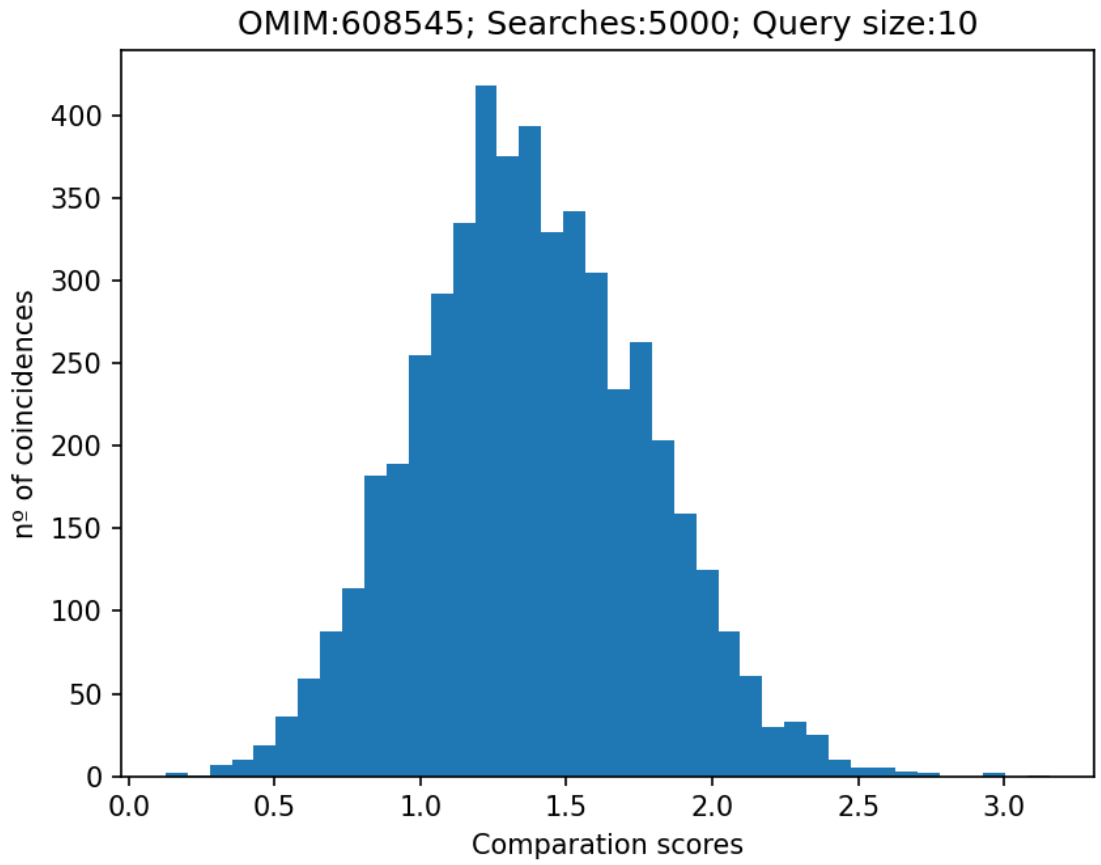


Figure 8 Histogram of the similarity scores for disease OMIM:608545 (Larsen-Like syndrome) with 5000 searches and query size of 10

Figure 8 presents a more irregular shape than Figure 7 due to the lower number of samples. Next, to visualize the differences better, a table showing the scores corresponding to the p-values pattern is presented.

p-values pattern	***Scores 100000***	***Scores 5000***
1	0.011425085	0.12576976
0.9	0.8760333	0.87632257
0.8	1.0412315	1.0460843
0.7	1.1635314	1.1668986
0.6	1.2696304	1.2605296
0.5	1.3703285	1.3595504

0.4	1.4720435	1.4608927
0.3	1.5819643	1.5768675
0.2	1.7135792	1.7198175
0.1	1.8967872	1.8921679
0.09	1.9209521	1.9137243
0.08	1.94914	1.938979
0.07	1.9769826	1.9645795
0.06	2.0118568	1.9979156
0.05	2.051991	2.0319717
0.04	2.0961797	2.0741363
0.03	2.149192	2.1213963
0.02	2.223653	2.2020078
0.01	2.3438337	2.3292305
0.005	2.4608805	2.4251823
0.001	2.706848	2.688992
0	3.2247465	3.1565912

Table 2. Scores comparison between 100000 and 5000 samples for Larsen-Like syndrome

Histograms of 5000 and 100000 samples are very similar in this case. Obviously, 5000 samples make the histogram a little less accurate, but looking at Table 2, the differences in the scores pattern are low. Due to the greater number of samples in 100000, the maximum and minimum values are more extreme that in the 5000 samples pattern. This won't affect the final results very much, as will be proved in chapter 4. More examples in the annex, section *Scores comparisons between 100000 and 5000 samples*. Reference to code is in annex section: *Code Index.*

1.10. Simulation of the patients

To validate this diagnostic algorithm, it needs to be tested in patients with realistic symptoms. However, according to Köhler et al. 2009, is very difficult to use real patients because getting phenotypic information from hundreds or thousands of patients, using standardized vocabulary and procedures, is infeasible without a lot of resources and time. This is why an informatic approach is used. They identified 44 complex dysmorphology syndromes for which adequate frequency data were available. Is assumed that the presence of individual symptoms is independent of others, which is incorrect, but not enough data is available to do otherwise.

1.10.1. Dysmorphology syndromes data

Before starting to process the 44 syndromes to generate simulated patients, some considerations have to be done before about the syntaxis of the data.

Example syndromes

Symptom name	Symptom ID	Probability of appearance
Atrophic skin and Aplasia cutis congenita and Scarring	HP:0001077 and HP:0001057 and HP:0000987	57.00%
Myopia and/or Ptosis and/or Nystagmus	HP:0000545 and/or HP:0000508 and/or HP:0000639	57.00%
Microphthalmos or Anophthalmia	HP:0000568 or HP:0000528	44.00%
Hypoplastic superior	HP:0008559	43.00%
Female: Intrauterine growth retardation	HP:0001511	14/22

Table 3. Some random symptoms selected (12)

In Table 3, there is an example of some symptoms extracted from the supplemental data (12) provided in the paper. They specified the probabilities of occurrence of each symptom, but some of them come in sets. In the example, the set of symptoms *Atrophic skin and Aplasia cutis congenita and Scarring*, have a 57% of chance of appearing in all, while *Microphthalmos or Anophthalmia* have a 44% of chance of appearing one or the other exclusively. The *and/or* conjunction used *in Myopia and/or Ptosis and/or*

Nystagmus means that one, two, or all symptoms can appear at once with a 57% of chance. The number of symptoms that intervene in the *and/or* case is chosen randomly.

```
ArrayList<String> HP_IDS_text = /*Array with Symptoms ids like: HP:0002046,
HP:0000651, etc*/

//Select randomly from 1 to all symptoms from HP_IDS_text
int numOfChoices = rand.nextInt(HP_IDS_text.size()) + 1;
Set<String> choices = new HashSet<String>(); //Results set

while(choices.size() < numOfChoices) {
    int choice = rand.nextInt(HP_IDS_text.size());
    choices.add(HP_IDS_text.get(choice));
}
```

Code 3.8 "and/or" processing

On the other hand, *or* implementation is straightforward, just choosing one symptom randomly.

```
ArrayList<String> HP_IDS_text = /*Array with Symptoms ids like: HP:0002046,
HP:0000651, etc*/

Set<String> choices = new HashSet<String>(); //Results set
int choice = rand.nextInt(HP_IDS_text.size());
String HPO = HP_IDS_text.get(choice);
choices.add(HPO);
```

Code 3.9 "or" processing

Finally, *and* will return all symptoms of the set as is.

```
ArrayList<String> HP_IDS_text = /*Array with Symptoms ids like: HP:0002046,
HP:0000651, etc*/

Set<String> choices = new HashSet<String>(HP_IDS_text); //Results set
```

Code 3.10 "and" processing

1.10.2. Patients' generation

For each of the 44 complex dysmorphology syndromes, 100 patients will be generated considering the probabilities of developing each symptom, obtaining a total of 4400 simulated patients. Every patient is assigned as male or female randomly, because some symptoms are gender-specific, like *Intrauterine growth retardation* in Table 3 which is specific to females. An example will be used to better illustrate this, using Table 3 as it were real disease symptoms:

The genre is randomly decided, in this case, male.

- The first symptom set *Atrophic skin and Aplasia cutis congenita and Scarring* have a 57% chance of appearing. If a random number that can go from 0.0 to 1.0 generates a number greater than 0.57, then this symptom is added considering the restrictions exposed in 3.4.1 Dysmorphology syndromes data.
- When the last symptom in Table 3 is reached (*Female: Intrauterine growth retardation*), it is ignored, because is exclusive to females.

The code showing this behavior is presented:

```
Set<Symptom> diseaseSymptoms; /*
Contains all symptoms of one of the 44 diseases. Every symptom includes the
probabilities, name,
and the relation with other symptoms utilizing an "and", "or", or "and/or"
*/

Random rand = new Random();

//Undefined = 0; Male = 1; Female = 2
int genre = 1;//genre decided for the patient, this is random

//The symptoms included in the generated patient are stored here
Set<String> selectedSymptoms = new HashSet<>();

for(Symptom symptom : diseaseSymptoms) {
    //Here is checked if the symptoms have genre constrains
    if(symptom.getGenre() == 0 || symptom.getGenre() == genre) {
        //if the chance of one symptom to happen is 57%, if the random
        //float generators goes above 0.57, the symptom is added to
selectedSymptoms
        if(symptom.getProbabilities() >= rand.nextFloat()) {
            selectedSymptoms.addAll(new HashSet<>(tempSymptomsArray));
        }
    }
}
```

Code 3.11 Patients' symptoms selection process implementation

1.10.3. Noise and imprecision

Real patients can have not only symptoms from one single disease, but they can also have some related to an underlying disorder that has nothing to do with the main problem. In Köhler et al. 2009 this is called "noise". To simulate noise, they added half as many noise terms to the terms selected from the underlying disease. If the disease has nine features, four randomly chosen terms were added. The noise symptoms won't be ancestors or descendants of the terms annotated to the disease or of each other.

```
Set<String> symptoms = //symptoms of the patient;
Set<URI> allSymptomsFromDb = //allSymptoms in HPO database descendants of
Phenotypic abnormality (HP:0000118)

int sym_size = symptoms.size();
int noise_size = sym_size / 2;

//Make a copy to avoid problems
ArrayList<URI> allSym = new ArrayList<>(allSymptomsFromDb.size());
allSym.addAll(allSymptomsFromDb);

//Create a set of all symptoms from this patient plus descendants and ancestors
//to avoid using noise that is contained in this set
Set<URI> ancAndDesc = new HashSet<>();
for(String sym : symptoms) {
    URI symURI = Load_Diseases.getUriFromHP_ID(factory, sym);
    ancAndDesc.addAll(engine.getDescendantsInc(symURI));
    ancAndDesc.addAll(engine.getAncestorsInc(symURI));
}

//Now this set contains only unrelated symptoms with the patient's disease
allSym.removeAll(ancAndDesc);

//Random symptoms from allSym are selected until noise_size is fullfilled
Set<URI> noise = new HashSet<>();

while(noise.size() < noise_size) {
    int choice = rand.nextInt(allSym.size());
    URI select = allSym.get(choice);

    noise.add(select);
}

//Finally, noise is added to the symptoms of the patient
for(URI HPO : noise) {
    String hp_id = Load_Diseases.getHP_IDFromUri(factory, HPO);
    symptoms.add(hp_id);
}
```

Code 3.12 Noise implementation

Physicians may not choose the appropriate words for describing a symptom that is present in the HPO database. Sometimes they describe a clinical anomaly at a more general level because clinical investigations have yet to be performed, or because they are unaware of more specific and correct terminology. This is called “imprecision”. When imprecision mode is used, every symptom of the patient is replaced randomly by one of its ancestors except the root symptom of *Phenotypic abnormality*.

If both “noise” and “imprecision” are applied, the first imprecision is used because it may lead to a reduced number of symptoms. This is because two terms can have the same ancestors. After this, “noise” is applied.

```
Set<String> symptoms = //symptoms of the patient;

if(addImprecision) {
    Set<String> imprecision = new HashSet<>();
    for(String symURI : symptoms) {

        //Get all ascendants of symptom
        Set<URI> ascendants = new HashSet<>();
        ascendants.addAll(engine.getAncestorsInc(symURI));

        //Remove root of ontology and virtualroot to avoid selecting it
        Set<URI> temp =
engine.getAncestorsInc(factory.getURI("http://hp/0000118"));
        ascendants.removeAll(temp);

        //Choose some random ancestor from "ascendants"
        int choice = rand.nextInt(ascendants.size());
        int counter = 0;
        for(URI asc : ascendants) {
            if(counter == choice) {
                imprecision.add(Load_Diseases.getHP_IDFromUri(factory, asc));
                break;
            }
            counter++;
        }

    }

    //Set patient's symptoms with the new imprecision set
    symptoms = imprecision
}

//If noise must be added now is the time
if(addNoise) {
    //Imprecision must be applied first because it may lead to a reduced number
of features.
    addNoise(engine, allSymptomsFromDb, factory, rand);
}
```

Code 3.13 Imprecision implementation

1.10.4. Benjamini and Hochberg correction

After the patients are generated, similarity scores are calculated along with the p-values. Each patient contains the p-values calculated for each disease, where the correct diagnosis should be the illness

with a lower p-value. In case of a draw, the disease with a higher similarity score will be selected. If the similarity scores are also equal, the best one will be randomly selected.

Because every patient contains thousands of p-values, some a priori good results could happen just by chance. To reduce the impact of this false positive, the Benjamini and Hochberg correction method is used.

$$P_{(i)}^{BH} = min\left(P_{(i)}\frac{m}{i}, P_{(i+1)}^{BH}\right) \quad \textbf{(Equation 3)}$$

Where $P_{(1)}$, $P_{(2)}$, …, $P_{(m)}$ are the p-values in ascendant order, and "m" is the number of values.

```
//The p-values in ascendant order
ArrayList<Entry<String, Float>> pValueList;

//The Benjamini and Hochberg equation is applied
int m = pValueList.size();
for (int i = m - 2; i >= 0; i--) {
    float pvalue = pValueList.get(i).getValue();
    float lastpvalue = pValueList.get(i+1).getValue();
    pValueList.get(i).setValue(Math.min(pvalue * m / (i + 1), lastpvalue));
}

return pValueList;
```

Code 3.14 Benjamini and Hochberg implementation

To illustrate the behavior of the Benjamini and Hochberg equation, an example will be provided using a set of ordered p-values and the corrected column. The last column is the alpha value. In Köhler et al. 2009, the p-values lower than 0.05 will be considered significative, in this example, 0.25 will be used for convenience. Reference to code is in annex section: *Code Index.*

p-value	p-value corrected	α = 0.25
0.001	0.0250000	ACCEPTED
0.008	0.1000000	ACCEPTED
0.039	0.2100000	ACCEPTED
0.041	0.2100000	ACCEPTED
0.042	0.2100000	ACCEPTED
0.061	0.2541667	REJECTED

0.074	0.2642857	REJECTED
0.205	0.4910714	REJECTED
0.212	0.4910714	REJECTED
0.216	0.4910714	REJECTED
0.222	0.4910714	REJECTED
0.251	0.4910714	REJECTED
0.269	0.4910714	REJECTED
0.275	0.4910714	REJECTED
0.340	0.5328125	REJECTED
0.341	0.5328125	REJECTED
0.384	0.5647059	REJECTED
0.569	0.7815789	REJECTED
0.594	0.7815789	REJECTED
0.696	0.8700000	REJECTED
0.762	0.9071429	REJECTED
0.940	0.9860000	REJECTED
0.942	0.9860000	REJECTED
0.975	0.9860000	REJECTED
0.986	0.9860000	REJECTED

Table 4. Benjamini and Hochberg example

Results

The main goal of this project was to compare different semantic similarity methods apart from the one used in the paper.

The first step is to analyze the performance of the semantic similarity method used in this thesis and compare it with the results shown in Köhler et al. 2009.

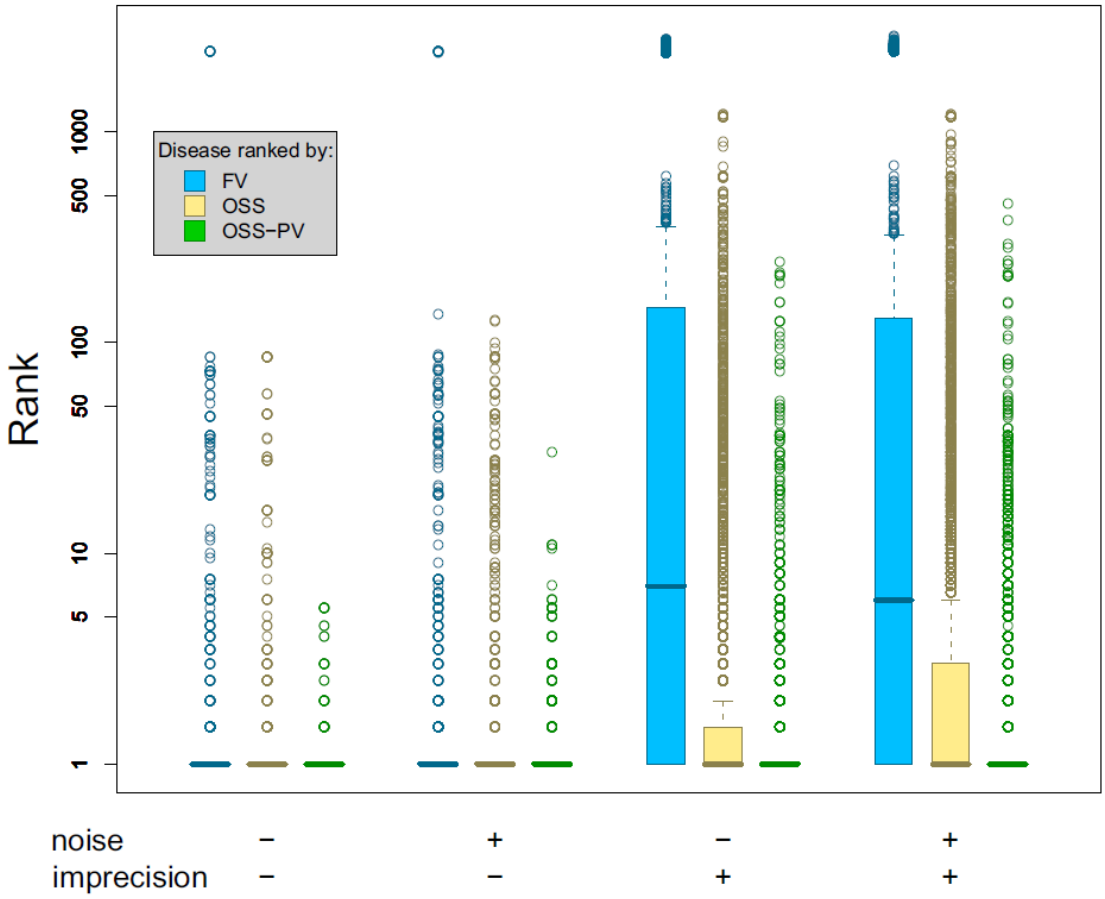


Figure 9. Results from Köhler et al. 2009.

In Figure 9, which is directly extracted from Köhler et al. 2009, the rankings of the real diseases are presented. There are 3 methods represented: the feature vector method (FV), the similarity scores method (3.2.3 Similarity score calculation) or OSS, and the p-value method (3.3 P-values calculation) or OSS-PV. The FV method just counts the number of symptoms of the patient that are present at each disease, then the higher score is considered the correct answer. Each set of methods is presented with different combinations of noise (randomly chosen terms) and imprecision (terms replaced by its

ancestors). Those ranks come from the list of diseases presented for each patient, if the correct disease is classified as rank 1, the result is perfect, while if the real disease is listed in the fourth position, the rank would be 4. For example, for the p-value method without noise and imprecision, the ranking shows that the vast majority of patients' diseases have been correctly identified and set as rank 1 except 7 patients (green dots).

Each boxplot shows 50% of the data surrounding the median line, from the first quartile to the third quartile. The whiskers extend from the box no more than 1.5 times the inter-quartile range, or in other words, the length of the box. The extreme points represent the data outside the end of the whiskers.

Figure 9, in conclusion, shows the overwhelming superiority of the OSS-PV method compared with the other two.

1.11. Results of the system implemented in this thesis and comparisons

For this first simulation, the most recent diseases database, as well as the most recent HPO database, were used. The graphics format is not the same as in Figure 9 for convenience. The same 4400 simulation patients are used in every method, 1100 in each combination of noise and imprecision.

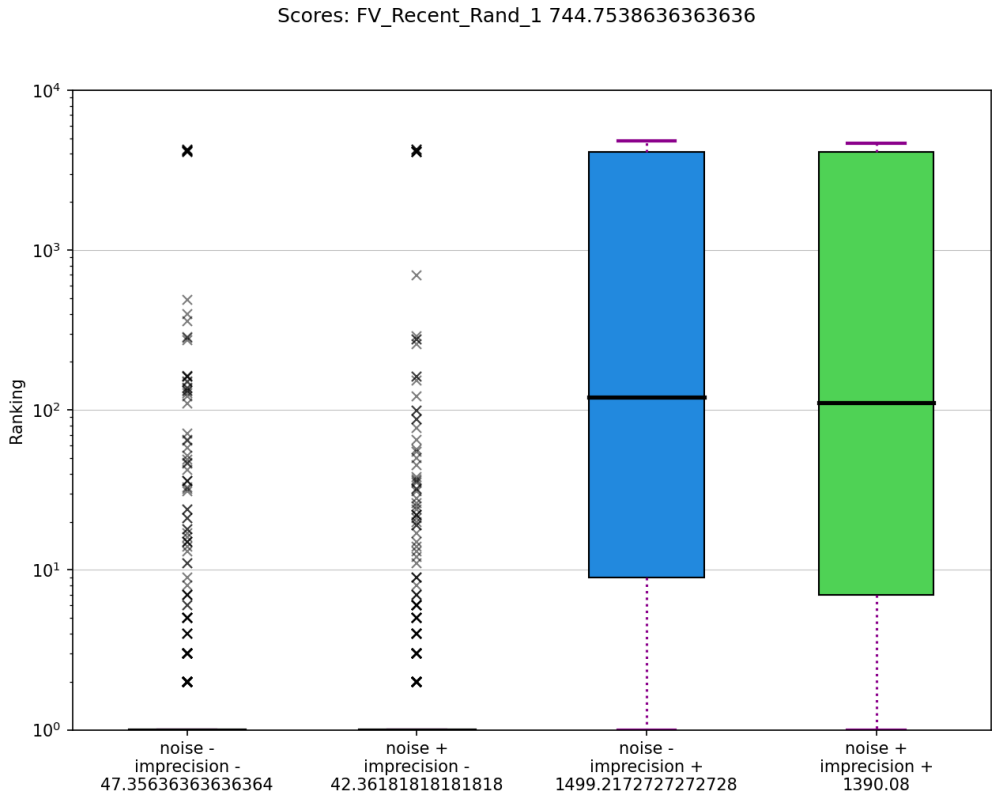


Figure 10. FV method ranking using recent databases

	Nº of 1st ranks	Medians
Noise - **Imprecision -**	969	1
Noise + **Imprecision -**	973	1
Noise - **Imprecision +**	112	120.5
Noise + **Imprecision +**	141	110.5

Table 4. Figure 10 statistics

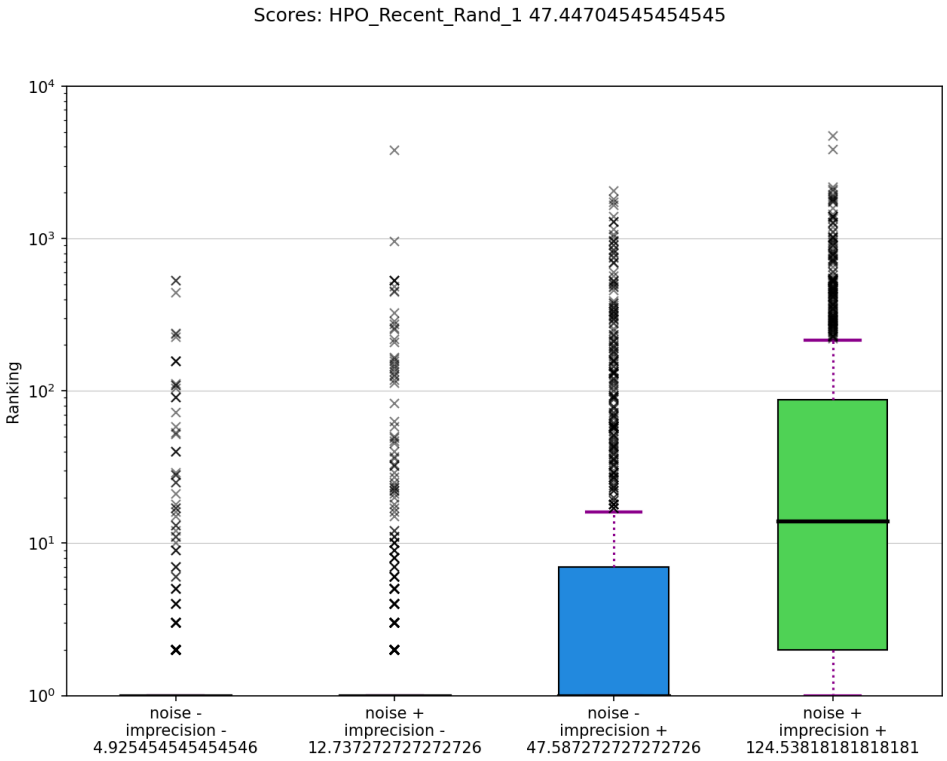


Figure 11. Similarity scores method ranking using recent databases

	Nº of 1st ranks	Medians
Noise - **Imprecision -**	981	1
Noise + **Imprecision -**	889	1
Noise -	623	1

Imprecision +		
Noise + **Imprecision +**	199	14

Table 5. Figure 11 statistics

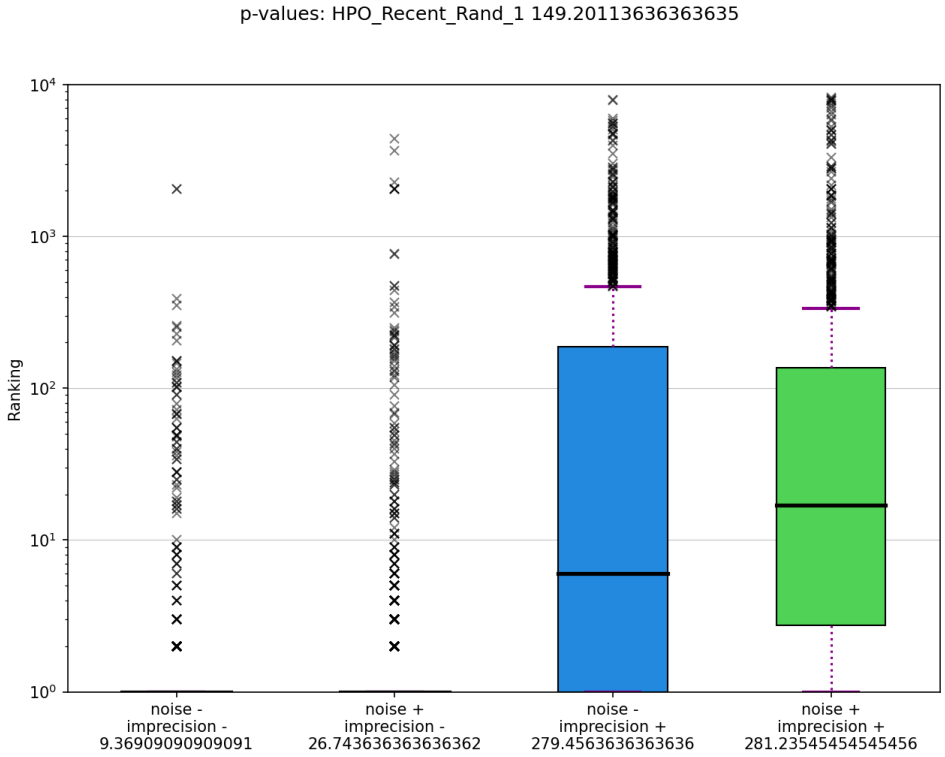


Figure 12. P-Values method ranking using recent databases

	Nº of 1st ranks	**Medians**
Noise - **Imprecision -**	968	1
Noise + **Imprecision -**	866	1
Noise - **Imprecision +**	391	6
Noise + **Imprecision +**	189	17

Table 6. Figure 12 statistics

Figures 10, 11, and 12 are divided accordingly to the presence of noise and imprecision, just as in Figure 9. Each figure represents one method: FV, Similarity scores, and P-values respectively.

The scores below each boxplot are just the ranks averaged. If, for example, one boxplot contains 1100 ranks which are 1, the average would be 1 as well; a perfect score. Obviously, that means that higher values are worse. The score in the title is the average of all samples including all variations of noise and imprecision. The similarity score method seems to give a better number of 1st ranks than using p-values.

Looking at the FV method (Figure 10) of the thesis compared with the results shown in the paper (Figure 9), the thesis simulation performs worse than the paper. The implementation of the FV method is very simple:

```
public double compare(Set<URI> query, Set<URI> disease, SM_Engine rc, SMconf
groupwiseconf) throws SLIB_Exception {

    //FV score
    int fv = 0;
    // Query terms are compared with disease symptoms, if there is a coincidence
    // f vis incremented
    for(URI a : query) {
        for(URI b : disease) {
            if(a.equals(b)) {
                fv++;
                break;
            }
        }
    }

    return fv;
}
```

Code 4.1 FV implementation

The patients were generated just as exposed in Köhler et al. 2009 (1), so the main causes for these discrepancies can be two:

- The paper didn't provide some key information about the implementation of the patients' simulation system
- The databases used in the thesis simulation are the cause of the bad performance

Another discrepancy found is that the similarity score system (Figure 11) provides better performance (the rank average scores are lower) than by p-value (Figure 12). Also, the "n*oise– imprecision+"* in p-values have a rank score almost as high as the "*noise+ imprecision+*" but with a much higher bias, the boxplot is bigger.

It seems that imprecision impacts heavily on the behavior of the simulation. Observing Figure 9 (results from Köhler et al. 2009), noise doesn't affect very much the performance, just as in the simulation.

Maybe there is some detail not explained in the paper about the imprecision, but due to the bigger impact on the results, a capped version of the imprecision method will be used to see if there is any improvement.

This modification consists of capping the random selection of a symptom ancestor, with only selecting, for example, ancestors that are “parents” or “grandparents” of the term. In the following examples, some limits are chosen randomly to observe their effects. A limit of 2 means that an ancestor can be only 2 generations away from the symptom, in other words, only “parents” and “grandparents” (or the symptom itself) can be chosen as the maximum. A limit of 3 means that, only “parents”, “grandparents”, and “great grandfathers” can be chosen.

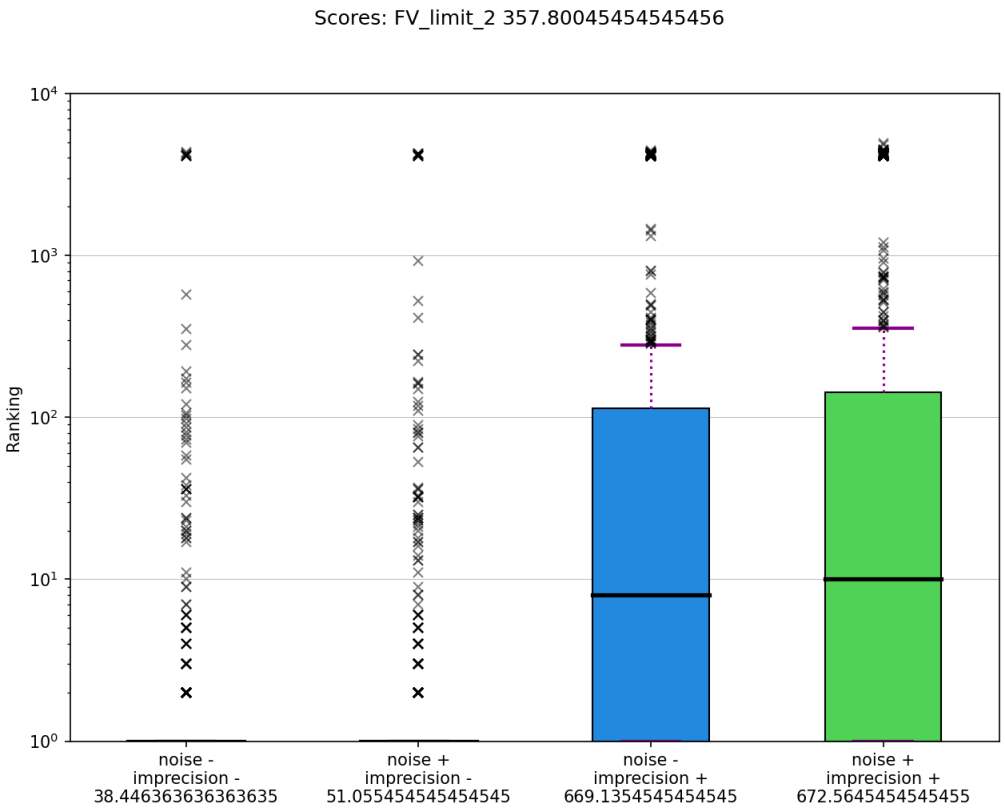


Figure 13. FV method with imprecision limit set to 2

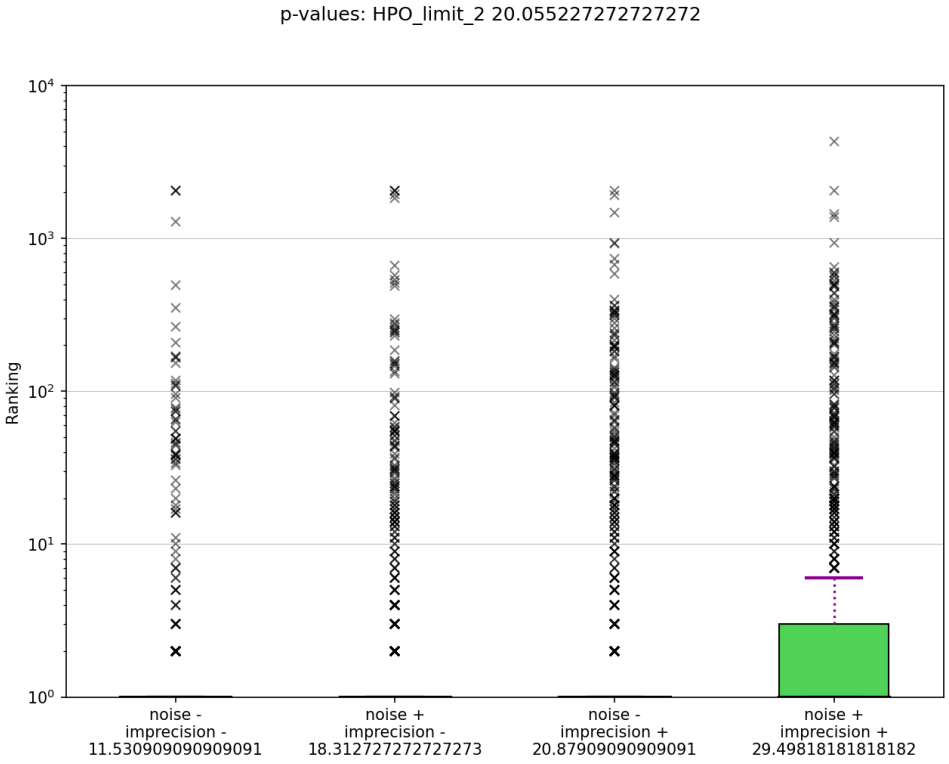


Figure 14. P-values method with imprecision limit set to 2

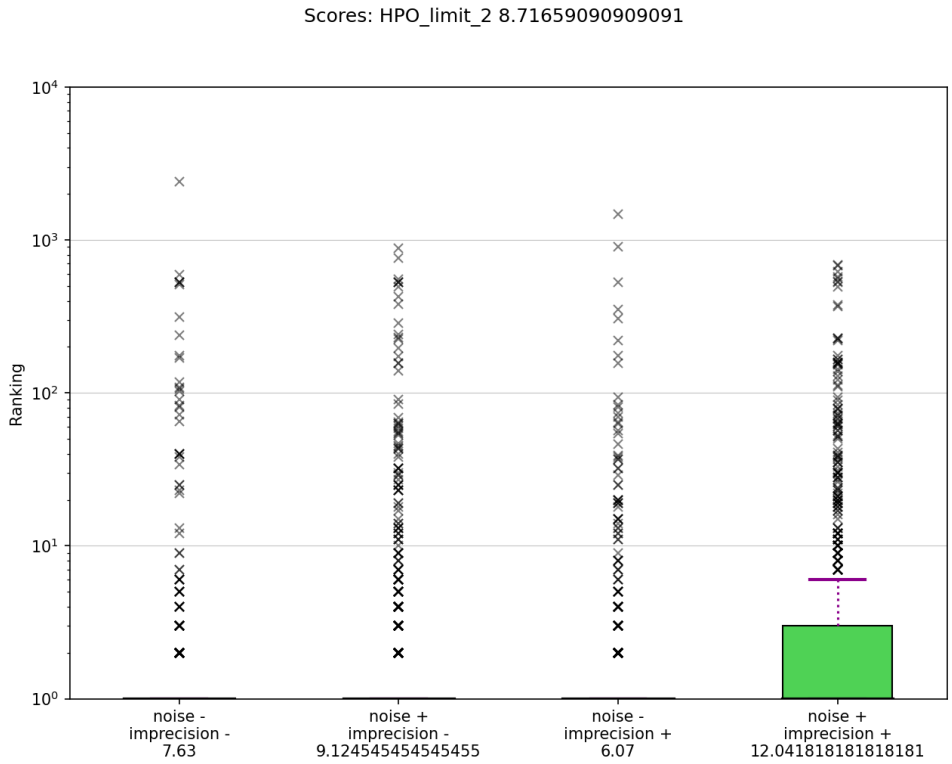
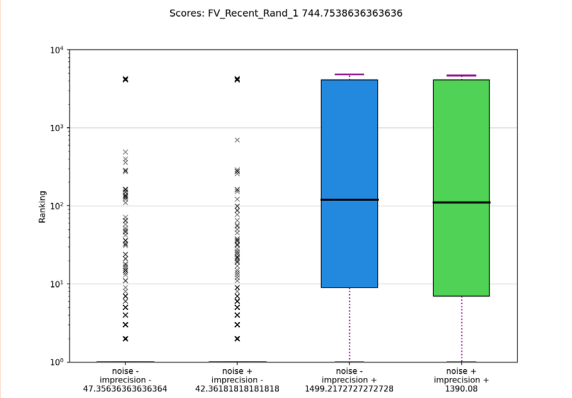
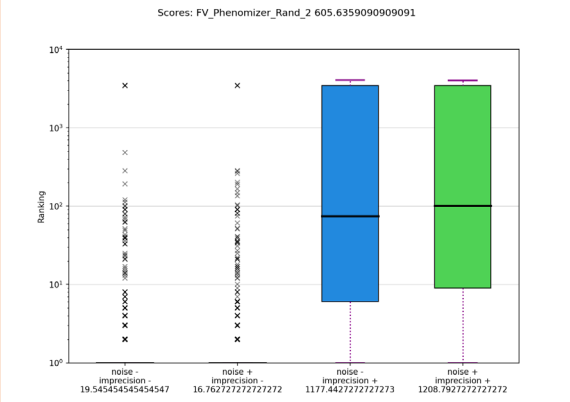


Figure 15. Similarity scores method with imprecision limit set to 2

Compared with Fig. 9, a limit of 2 still returns inconsistent results. While FV performance is very similar to Köhler et al. 2009, and the similarity score is better (Fig. 15), the p-value method is not as good. It can be concluded that the main problem of these inconsistencies, may not be related to the imprecision. More experiments with limits are in the annex.

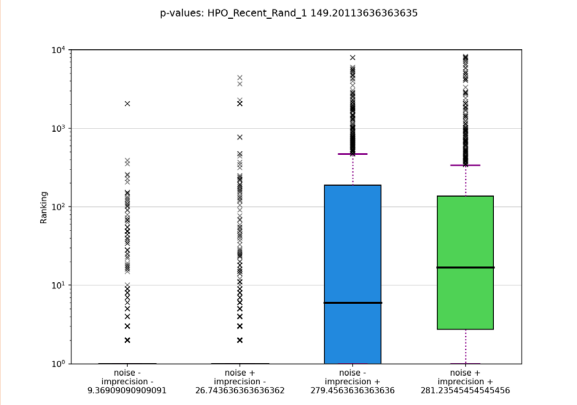
Another approach to address these problems is to use the HPO database version 1.59, similar to the one used in Köhler et al. 2009 (1.58), and a closer version in time of the annotation database which source will be explained in 4.2 Evaluation of Phenomizer using web scraping.

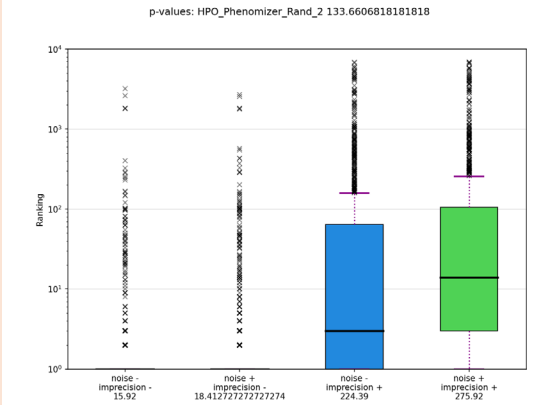
	Using the latest databases (25/05/2022)	**Using databases close in time with the paper**
FV		

	Nº of 1st ranks	**Medians**
Noise - Imprecision -	969	1
Noise + Imprecision -	973	1
Noise - Imprecision +	112	120.5
Noise + Imprecision +	141	110.5

	Nº of 1st ranks	**Medians**
Noise - Imprecision -	946	1
Noise + Imprecision -	951	1
Noise - Imprecision +	160	75
Noise + Imprecision +	145	101

P-values method

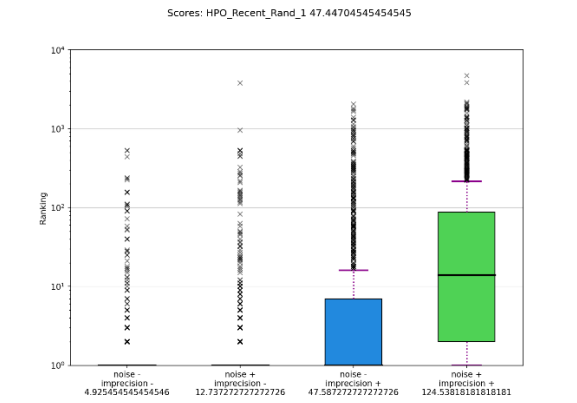


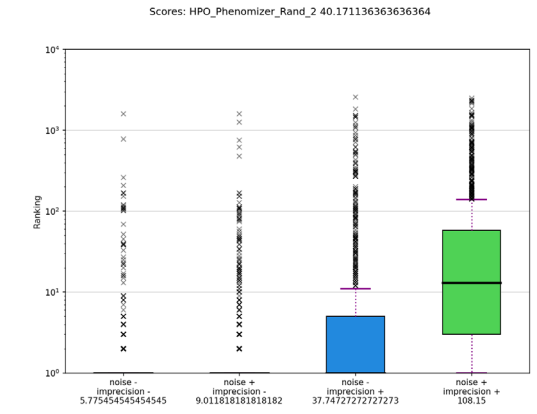


	Nº of 1st ranks	Medians
Noise - Imprecision -	968	1
Noise + Imprecision -	866	1
Noise - Imprecision +	391	6
Noise + Imprecision +	189	17

	Nº of 1st ranks	Medians
Noise - Imprecision -	933	1
Noise + Imprecision -	877	1
Noise - Imprecision +	467	3
Noise + Imprecision +	170	14

Similarity Scores





	Nº of 1st ranks	Medians
Noise - Imprecision -	981	1
Noise + Imprecision -	889	1
Noise - Imprecision +	623	1
Noise + Imprecision +	199	14

	Nº of 1st ranks	Medians
Noise - Imprecision -	963	1
Noise + Imprecision -	882	1
Noise - Imprecision +	668	1
Noise + Imprecision +	191	13

Table 7. Comparison using new and old databases

According to Table 7, using databases very similar in time to Köhler et al. 2009 improves the performance of the simulation. Also, the similarity score method performs better that the p-value method, mainly in the “noise- imprecision+” state. However, the performance is still worse than stated in the article. Another, more definitive, the approach will be used in the next section.

1.12. Evaluation of Phenomizer using web scraping

In Köhler et al. 2009 they implemented the system into an app called the *Phenomizer* (14). Comparing the simulation with the Phenomizer can be a good way of assessing if the results are similar in a “real” situation with the previously generated patients.

The idea is to generate patients like in previous sections and use them at the same time in the simulation developed in this project and the Phenomizer. To use the Phenomizer web scraping is required. Web scraping is a technique that consists of extracting data from websites, in this case from the Phenomizer itself.

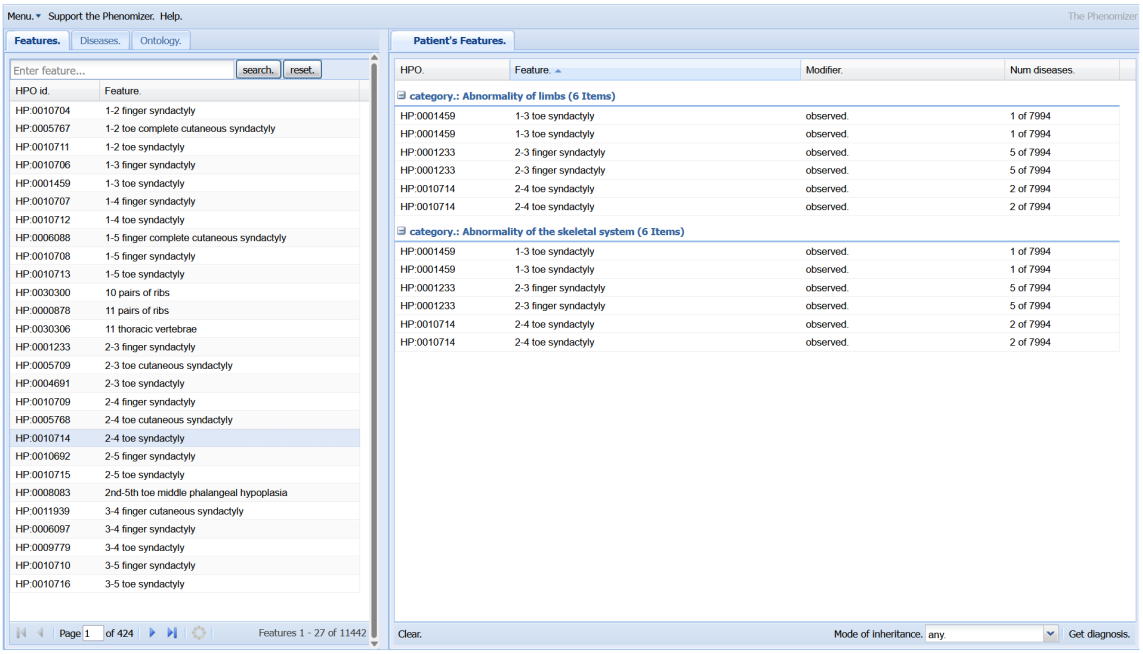


Figure 16. Phenomizer

4400 patients were generated, with noise and imprecision intercalated just as in section *3.4.2. patients’ generation*. This set of patients will be called *Set 0*. The code used for the web scraping is adjoined in the annex. The results comparing the Phenomizer performance with the simulation, using the latest databases, are the following:

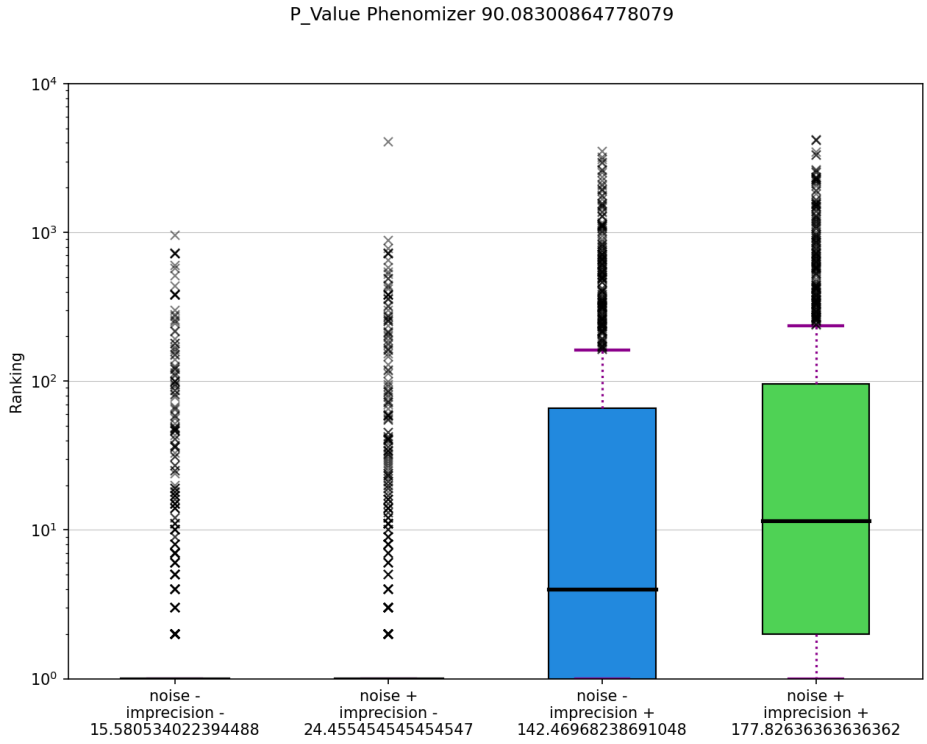


Figure 17. Phenomizer performance using the P-value method, the patient Set 0, and the latest databases

	Nº of 1st ranks	**Medians**
Noise - **Imprecision -**	939	1
Noise + **Imprecision -**	864	1
Noise - **Imprecision +**	335	4
Noise + **Imprecision +**	214	11.5

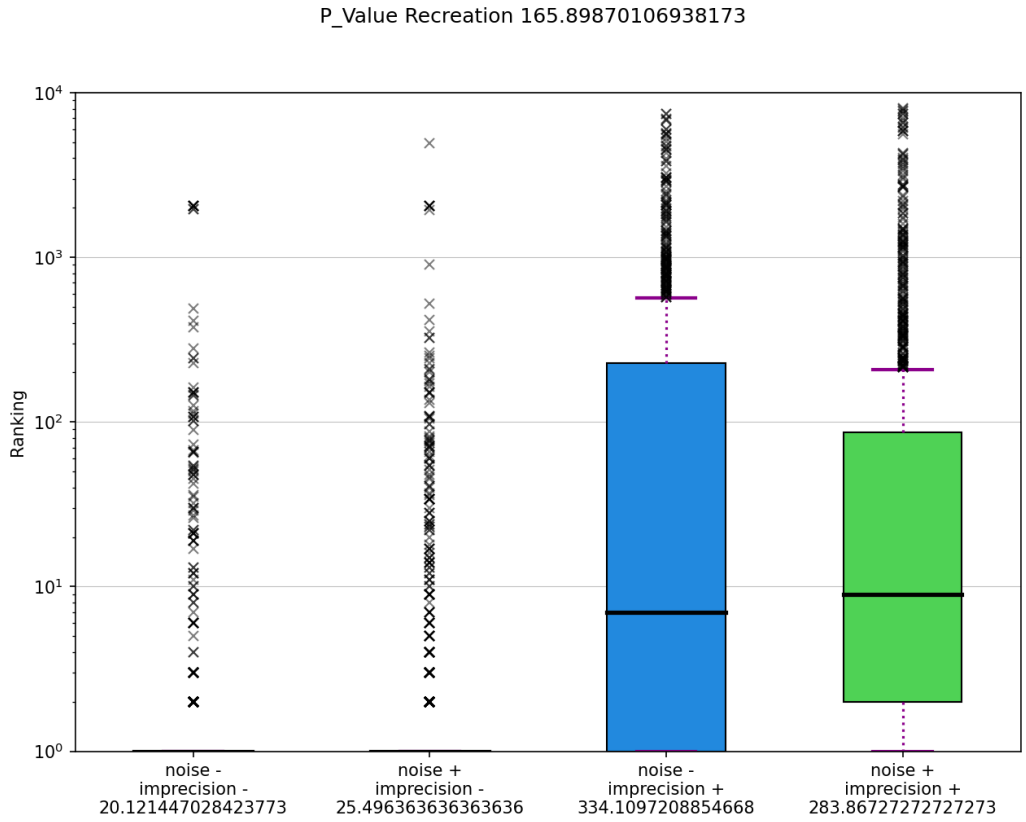


Figure 18. Local simulation performance using the P-value method, the patient Set 0, and the latest databases

	Nº of 1st ranks	**Medians**
Noise - Imprecision -	1010	1
Noise + Imprecision -	880	1
Noise - Imprecision +	351	7
Noise + Imprecision +	249	9

It can be seen that the performance of the local simulation and the Phenomizer are very close, however, there is a big bias in the local simulation in the “noise- imprecision+” boxplot. Also, the median in “noise+ imprecision+” is lower in the local simulation but the bias is clearly higher looking at the averages of both boxplots: 283.86 in the local simulation, and 177.82 in the Phenomizer. Again,

this is mostly caused by the new HPO database. The number of correctly classified diseases is higher in the local simulation.

The most interesting point is that the results shown in the Phenomizer boxplots (Fig. 17), are not even close to the results in Köhler et al. 2009 (Fig. 9). This may be caused by some difference in the real generation of the patients used in Fig. 9 and the description of that process that, a posteriori, was used and implemented in this simulation.

To try to improve a bit more the outcome of the simulation, the annotations database from the Phenomizer itself will be extracted using web scraping again. Code references are in the annex.

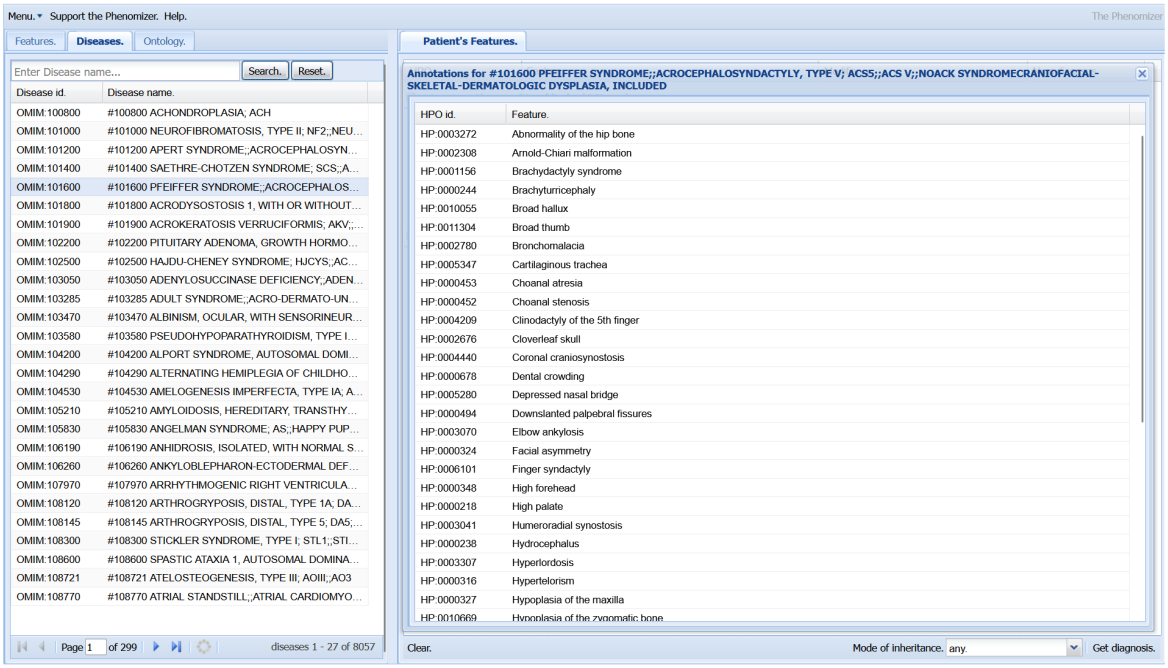


Figure 19. Phenomizer showing its annotations database

A new set of patients called *Set 1* was generated and the results, using the Phenomizer's annotations database and the HPO database version 1.59, are the following:

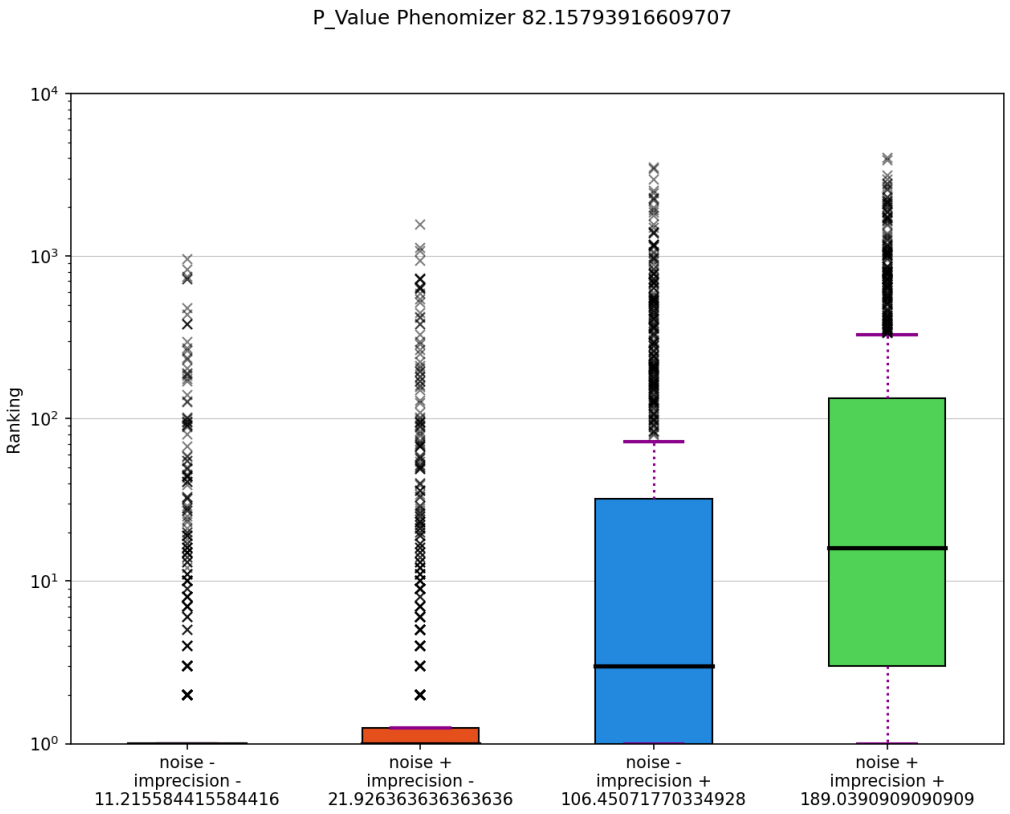


Figure 20. Phenomizer performance using the P-value method, the patients Set 1, and older databases

	Nº of 1st ranks	Medians
Noise - Imprecision -	956	1
Noise + Imprecision -	825	1
Noise - Imprecision +	398	3
Noise + Imprecision +	188	16

Table 8. Figure 20 statistics

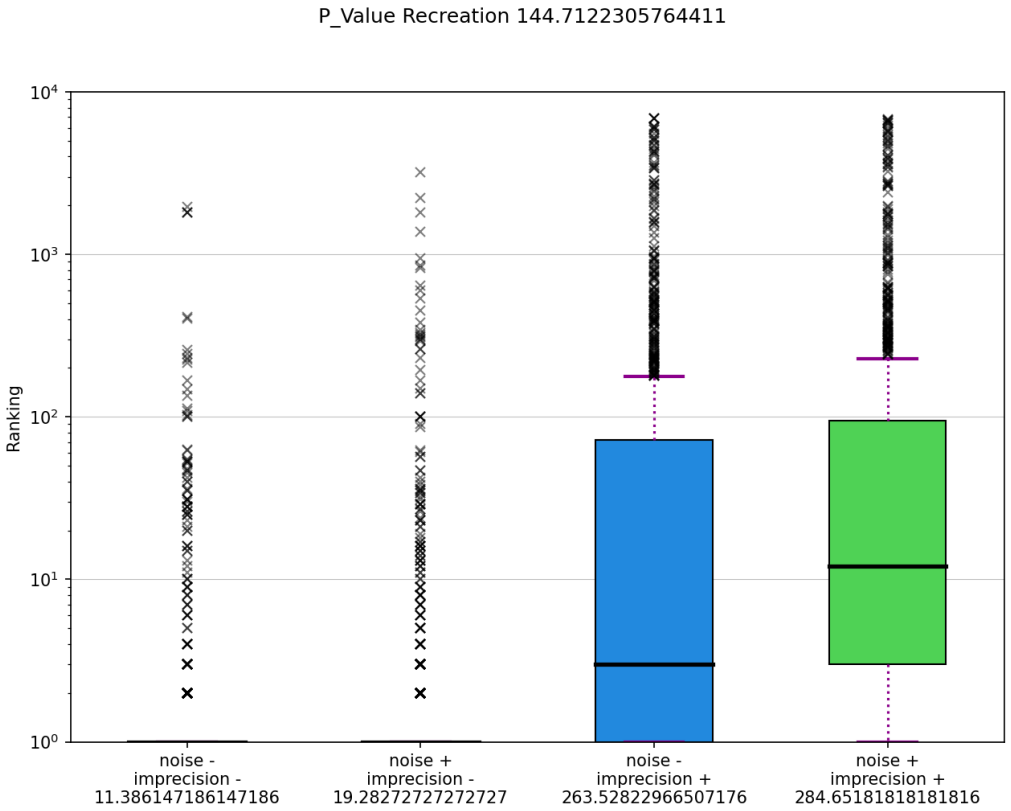


Figure 21. Local simulation performance using the P-value method, the patients Set 1, and older databases

	Nº of 1st ranks	**Medians**
Noise - Imprecision -	984	1
Noise + Imprecision -	860	1
Noise - Imprecision +	411	3
Noise + Imprecision +	194	12

Table 9. Figure 21 statistics

The local simulation and the Phenomizer results are very similar in terms of medians and bias. The medians are identical in Fig. 20 and Fig. 21 except for the boxplot at “noise+ imprecision+”, where the median is slightly lower in the local simulation.

Just as in the last example, the bias in “noise- imprecision+” is higher in the local simulation compared with the Phenomizer, but a bit lower in “noise+ imprecision+”. Still, the averages in the local simulation

are higher because there are more outliers with very high values that increase these numbers. However, these extreme outliers are not as important as the number of correctly classified diseases.

In conclusion in this section, the performance of the local simulation is in pair with the Phenomizer, and there are some improvements in “Noise- Imprecision+” using the older databases over the latest but worse results in “Noise+ Imprecision+”.

The p-value method, in fact, performs worse than with similarity scores, as seen in section *4.1 Results of the system implemented in this thesis and comparisons* and annex *4.5 Performance comparison between old and latest databases*. However, the p-values can contain valuable information about whether the disease in rank 1 is a reliable result (p-value < 0.05) or not (p-value > 0.05). This part will be discussed in the next section. Reference to code is in annex section: *Code Index.*

1.13. P-value reliability

It was proved in previous sections that the similarity scores method delivers better results in terms of bias and the number of correctly classified diseases, but the p-value method has the advantage of assessing if the results delivered are reliable or not. In this section, the quality of the p-values to tell if a value is reliable or not will be analyzed.

```
#Lists contain data from the 4400 auto-generated patients
real_rank_list = #List of ranks of the real diseases in the results
real_pvalue_list = #List of p-values of the real diseases
first_pvalue_list = #List of p-values of the first ranked diseases

ranktocompare = 1
significance = 0.05

FN = 0 #False Negatives
TP = 0 #True Positives
FP = 0 #False Positives
TN = 0 #True Negatives

for i in range(len(real_rank_list)):
    real_rank = real_rank_list[i]
    real_pvalue = real_pvalue_list[i]
    first_pvalue = first_pvalue_list[i]
    if real_rank <= ranktocompare:
        if(real_pvalue > significance):
            #If the real disease rank is 1, but the p-value is greater than 0.05
            #then is considered as a False Negative
            FN += 1
        else:
            TP += 1
    else:
```

```
        if(real_pvalue > significance):
            if(first_pvalue < significance):
                #If the real disease rank is greater than 1 and the first ranked
                #disease p-value is lower than 0.05, then is considered
                #as a False Positive
                FP += 1
            else:
                TN += 1
        else:
            #In case the real disease rank is higher than 1, but the p-value is
            #also lower than 0.05, then is difficult to choose
            #what disease is the correct answer and is considered a False
positive
            FP += 1
print("FN: " + str(FN))
print("TP: " + str(TP))
print("FP: " + str(FP))
print("TN: " + str(TN))
```

Code 4.2. Truth table implementation

To create the truth table, four variables are used: the significance value (in this case 0.05) to tell if the result is relevant, the rank of the real disease, the p-values of the 1st classified disease, and the p-value of the real disease.

To assess the quality of the p-values, a truth table will be used and a new set of patients will be generated for this goal.

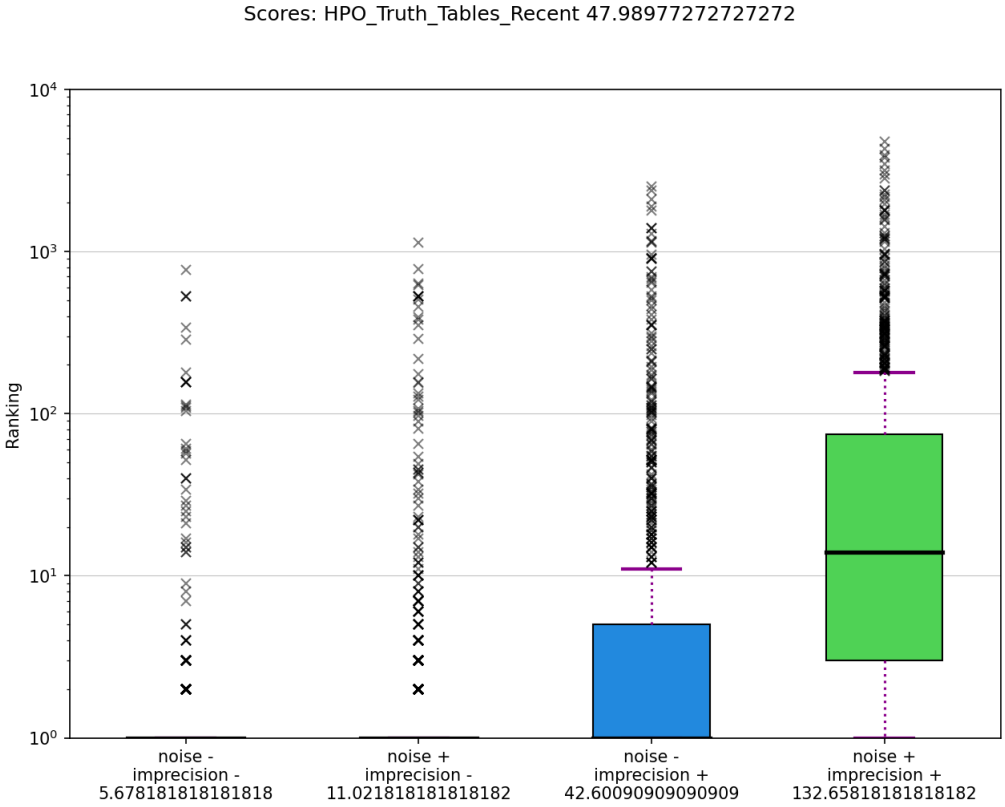


Figure 22. Local simulation using Similarity scores with the latest databases

	Nº of 1st ranks	**Medians**
Noise - Imprecision -	984	1
Noise + Imprecision -	891	1
Noise - Imprecision +	673	1
Noise + Imprecision +	172	14

Table 10. Figure 22 boxplot statistics

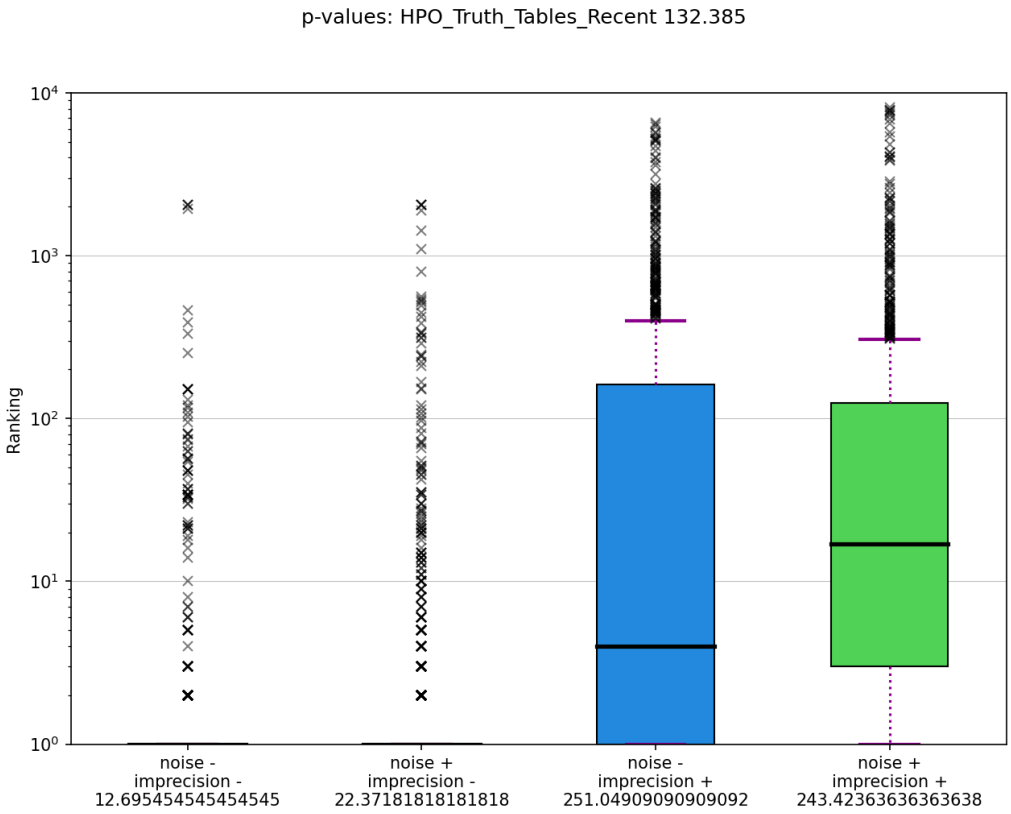


Figure 23. Local simulation using p-values with the latest databases

	Nº of 1st ranks	Medians
Noise - Imprecision -	969	1
Noise + Imprecision -	872	1
Noise - Imprecision +	411	4
Noise + Imprecision +	150	17

Table 11. Figure 23 boxplot statistics

Local simulation (latest databases) (Rank = 1)

	False Negative	True Positive	False Positive	True Negative
Noise – Imprecision -	0	969	124	7
Noise + Imprecision -	73	799	208	20
Noise – Imprecision +	305	106	378	311

Noise + Imprecision +	121	29	155	795
Total	499	1903	865	1133

Table 12. The truth table of Figure 23 for ranks equal to 1

Looking at Table 12, rows "Noise+ Imprecision+", and "Noise- Imprecision+"; which should be closer to a real case, some conclusions are extracted:

- Is very difficult to trust a positive result (p-value < 0.05), because only 29 out of 184 positives (15.7%) are really true. Without noise, the results improve a little to 106 out of 484 positives (21%), still not enough to be trusted.
- On the other hand, is much easier to trust a negative result: 795 out of 916 are true negatives (86.7 %), but "Noise- Imprecision+" is 311 out of 616 (50.4%).
- When the noise and imprecision of a set of symptoms are reduced, the ratio of true negatives decreases but the true positives ratio increases. This is because a simpler set of symptoms makes the p-value method more proficient in finding the correct disease.
- Compared with the stats of the similarity score (Table 10), in "Noise+ Imprecision+" 172 out of 1100 diseases (15.63%) are correctly classified, which is almost the same performance that with p-values. In "Noise- Imprecision+" 673 out of 1100 (61.18%), which is much better than the p-value method.

In terms of correctly identified diseases ratio, the Similarity score method is better than the p-value method. While is true that the ratio of true negatives for "Noise+ Imprecision+" is high enough, only 29 patients (2.6%) have been correctly diagnosed (172 with similarity scores), and 106 patients (9.6%) in "Noise- Imprecision+" (673 with similarity scores).

This is a tradeoff between a good indicator of true negatives and low true positive accuracy when the symptom set has imprecision and noise; and, improving the symptoms set through more diagnosis and investigations, lowering the true negative ratio abruptly while increasing slightly the number of true positives. Moreover, is not possible in a real scenario to tell which quantity of noise and imprecision your symptoms set has, in consequence, the exact ratios are unknown.

To make this analysis more complete, a new truth table from Fig. 23 is obtained but counts as a true positive if the real patient' disease is in the top ten. A patient's disease present in the top ten results can still be useful for a researcher to try more diagnosis techniques to narrow down this list.

Local simulation (latest databases) (Rank <= 10)

	False Negative	True Positive	False Positive	True Negative
Noise – Imprecision -	16	1021	59	4
Noise + Imprecision -	134	873	81	12
Noise – Imprecision +	520	112	273	195
Noise + Imprecision +	427	35	111	527
Total	1097	2041	524	738

Table 13. The truth table of Figure 23 for ranks equal to 10

"Noise+ Imprecision+" from table 13 shows a much higher number of false negatives. While the number of true positives in "Noise+ Imprecision+" and "Noise- Imprecision+" is very similar, the false negatives are more abundant, this means that a lot of correct diagnosed diseases present in the top ten were assigned a high p-value, which invalidates the result.

In conclusion, the p-values method is not very useful to diagnose the correct patient's disease due to the low proportion of true positives. Even if the researcher is only analyzing the first result and not the top ten, the uncertainty about the quantity of imprecision and noise in the symptoms can be misleading when looking for true negatives. Is better to just use the similarity scores method, with 15.63% of correctly diagnosed patients with high imprecision and noise. This percentage can improve very fast when "cleaning" the symptoms set.

1.14. More Similarity score methods

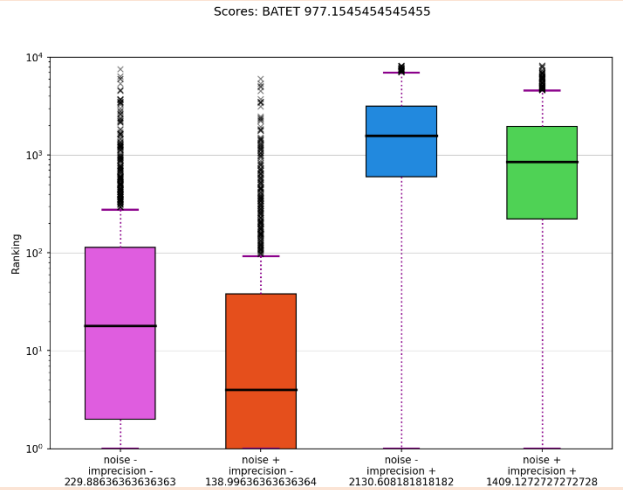
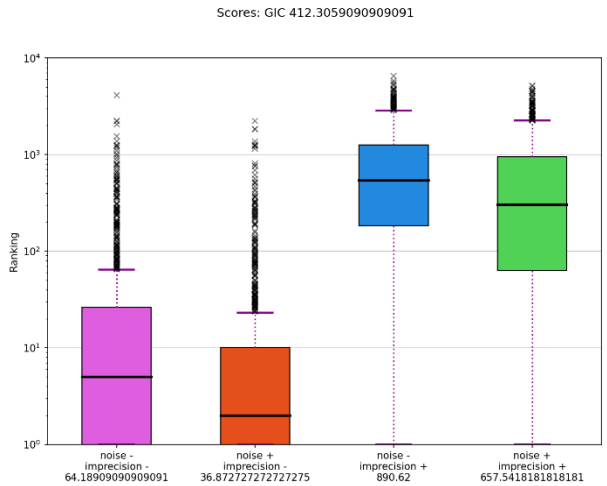
The last part of this thesis is to test more Similarity scores methods apart from Eq. 1. Maybe some new technique can achieve better performance than the method used in this thesis and Köhler et al. 2009. The new methods are extracted from the *Semantic Measures Library (SML)* (15). The library contains many score calculation methods divided into many subsections. In this case, the focus will be put on the next sections:

Direct Groupwise measures: This kind of similarity method normally compares two groups of terms (like query and symptoms) using set operations like intersection, or union of both groups, and obtains scores playing with the sizes of the sets.

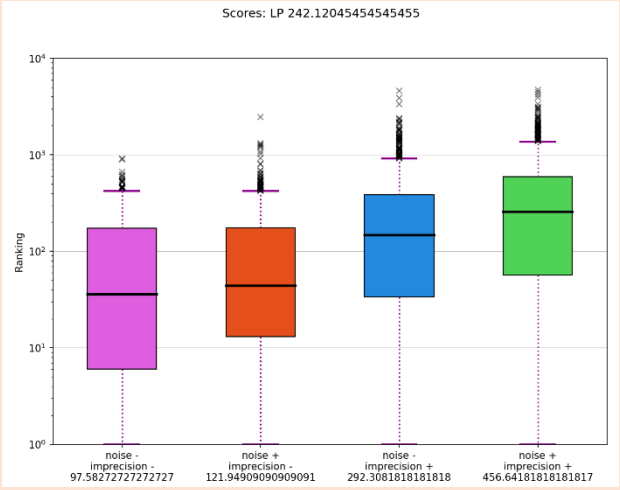
IC-based measures: This set of similarity methods is based into compare every item of one set with every item of the other using their IC values. If, for example, the first set *query* contains 5 items, and the second set *symptoms* contain 7 terms, a matrix called *matrix score* of size 7x5 will be created with every score of the pairwise comparisons. Then, this matrix is processed using an *Indirect Groupwise* measure like taking the max value from the matrix, an average, etc.

1.14.1. Direct Groupwise measures

The best performant direct Groupwise measures are shown in the next table. The full set of methods is in the annex.

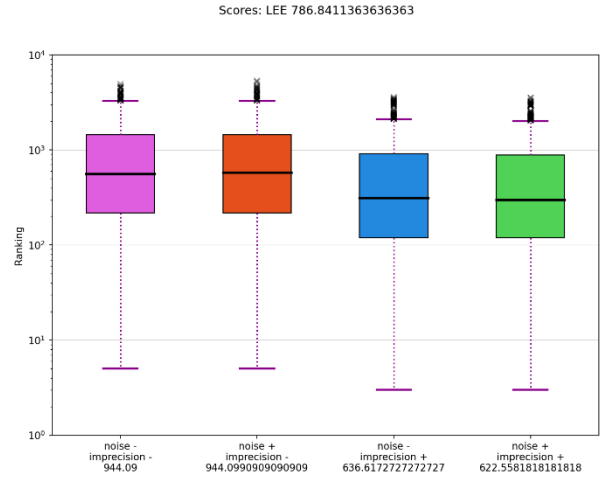
Method	Boxplot	Boxplot performance		
Batet 2010			Nº of 1st ranks	Medians
		Noise - Imprecision -	217	18.0
		Noise + Imprecision -	389	4.0
		Noise - Imprecision +	3	1578.0
		Noise + Imprecision +	5	854.5
Sim GIC			Nº of 1st ranks	Medians
		Noise - Imprecision -	325	5.0
		Noise + Imprecision -	526	2.0
		Noise - Imprecision +	5	539.5
		Noise + Imprecision +	16	304.5

SimLP



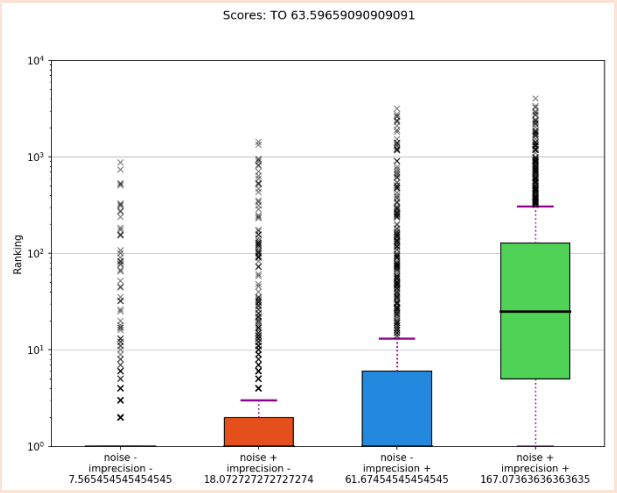
	Nº of 1st ranks	Medians
Noise - Imprecision -	124	36.0
Noise + Imprecision -	92	44.0
Noise - Imprecision +	28	147.0
Noise + Imprecision +	21	257.0

Lee 2004

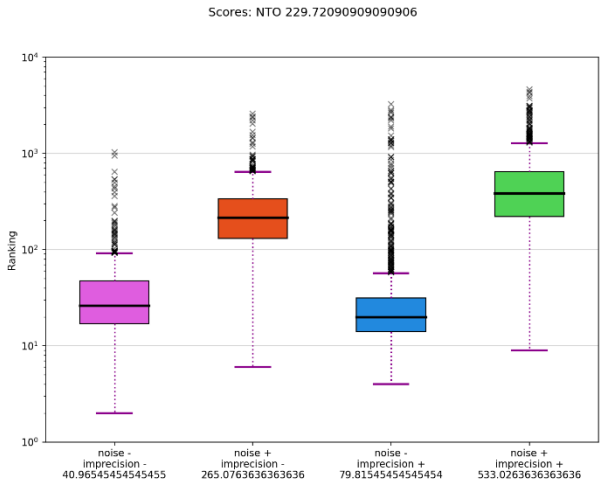


	Nº of 1st ranks	Medians
Noise - Imprecision -	0	561.0
Noise + Imprecision -	0	577.0
Noise - Imprecision +	0	314.0
Noise + Imprecision +	0	299.5

Term Overlap



	Nº of 1st ranks	Medians
Noise - Imprecision -	962	1.0
Noise + Imprecision -	748	1.0
Noise - Imprecision +	666	1.0
Noise + Imprecision +	126	25.0

Normalized Term Overlap			

	Nº of 1st ranks	Medians
Noise - Imprecision -	0	26.0
Noise + Imprecision -	0	215.0
Noise - Imprecision +	0	20.0
Noise + Imprecision +	0	385.0

Table 14. Best direct Groupwise measures by similarity scores using the latest databases

The best results are given by the Term Overlap method, based on counting the number of common ancestors from both sets.

	Nº of 1st ranks	Medians
Noise - Imprecision -	984	1
Noise + Imprecision -	891	1
Noise - Imprecision +	673	1
Noise + Imprecision +	172	14

Table 15. Performance statistics by similarity scores of Köhler et al. 2009 method

	Nº of 1st ranks	Medians
Noise - Imprecision -	962	1.0
Noise + Imprecision -	748	1.0
Noise - Imprecision +	666	1.0
Noise + Imprecision +	126	25.0

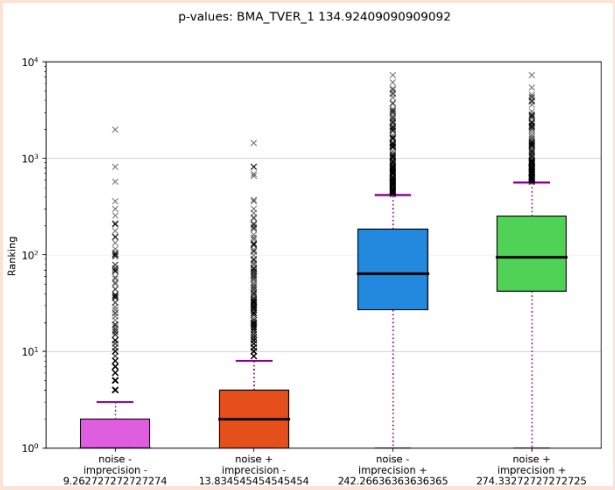
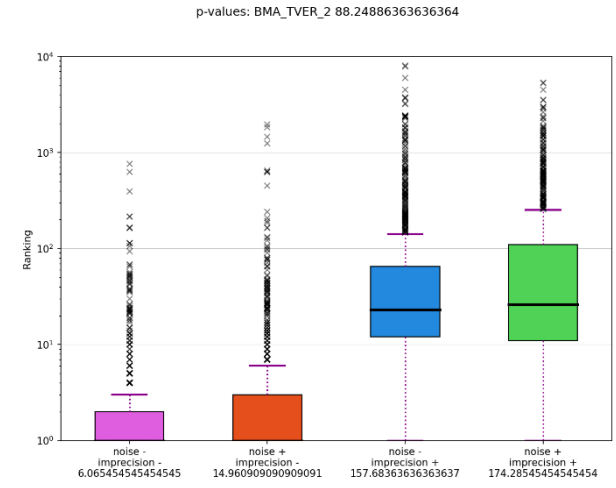
Table 16. Performance statistics by similarity scores of Term Overlap method

The term overlap method performs very similarly to Köhler et al. 2009 method (Eq 1) but without considering the IC values. The bias of Term Overlap is also higher. Even so, Köhler et al. 2009 method

using similarity scores are still better. There is no good candidate to substitute Köhler et al. 2009 in the direct Groupwise measures.

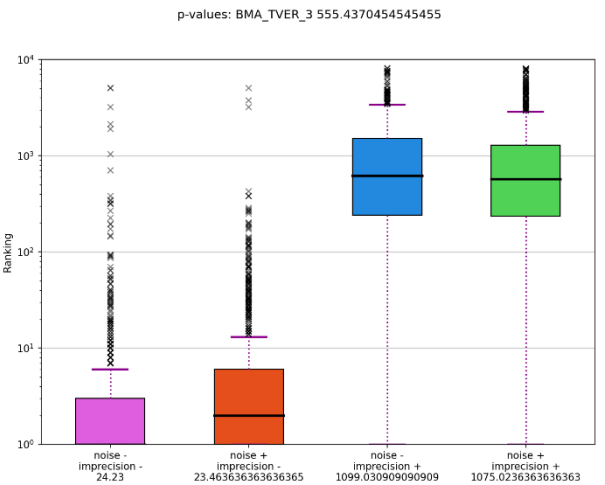
1.14.2. IC-based measures

A new table with IC-based measures and indirect measures is presented where only the best methods are included. The full table is adjoined in the annex.

Method	Boxplot	Boxplot performance		
Pairwise: Tversky IC Contrast model **Indirect measure:** Best Match Average (BMA)			**Nº of 1st ranks**	**Medians**
		Noise - Imprecision -	724	1.0
		Noise + Imprecision -	545	2.0
		Noise - Imprecision +	11	64.0
		Noise + Imprecision +	6	95.0
Pairwise: Tversky IC Ratio model **Indirect measure:** Best Match Average (BMA)			**Nº of 1st ranks**	**Medians**
		Noise - Imprecision -	763	1.0
		Noise + Imprecision -	624	1.0
		Noise - Imprecision +	27	23.0
		Noise + Imprecision +	12	26.0

Pairwise: Tversky IC Prop Contrast model

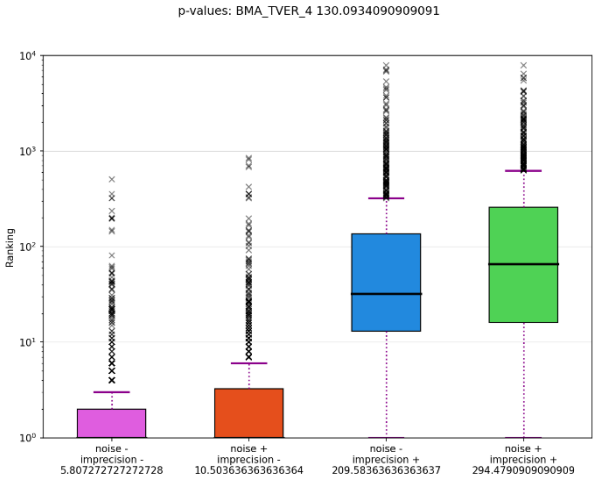
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	677	1.0
Noise + Imprecision -	411	2.0
Noise - Imprecision +	5	618.0
Noise + Imprecision +	3	576.0

Pairwise: Tversky IC Prop Ratio model

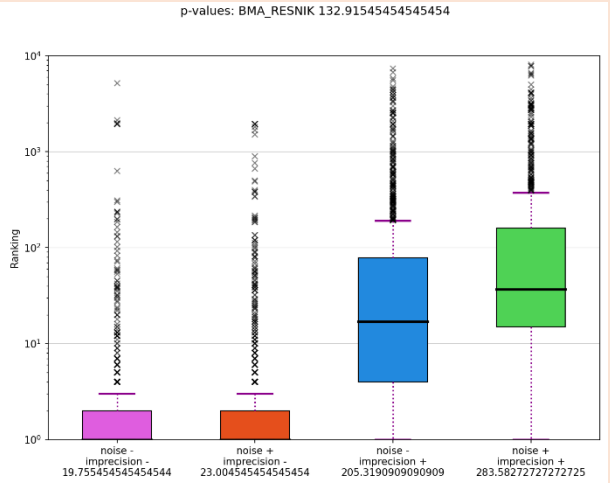
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	796	1.0
Noise + Imprecision -	565	1.0
Noise - Imprecision +	21	32.0
Noise + Imprecision +	15	66.0

Pairwise: Resnik 1995

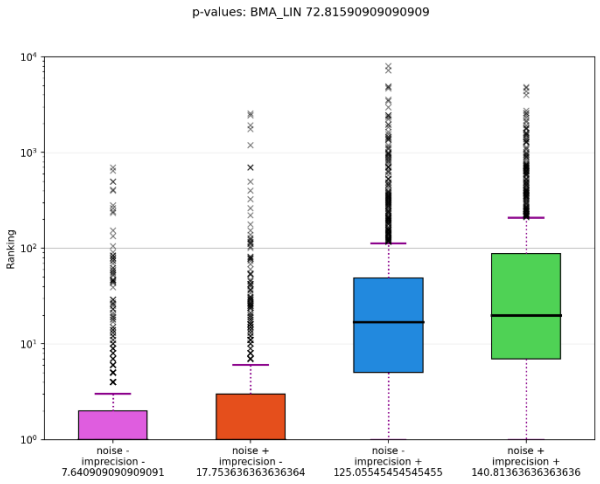
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	769	1.0
Noise + Imprecision -	686	1.0
Noise - Imprecision +	122	17.0
Noise + Imprecision +	55	37.0

Pairwise: Lin 1998

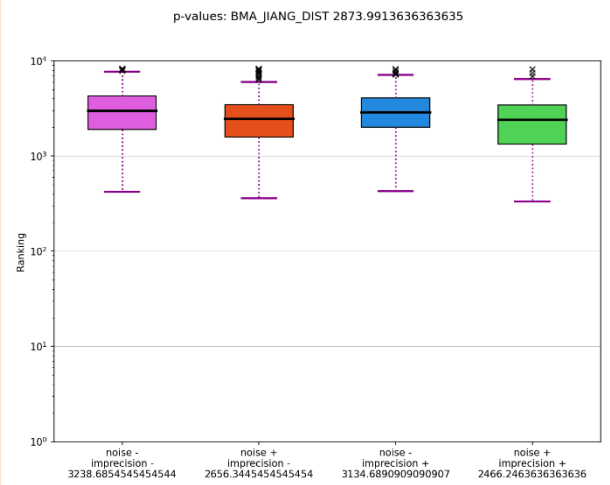
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	753	1.0
Noise + Imprecision -	660	1.0
Noise - Imprecision +	101	17.0
Noise + Imprecision +	83	20.0

Pairwise: Jiang and Conrath 1997

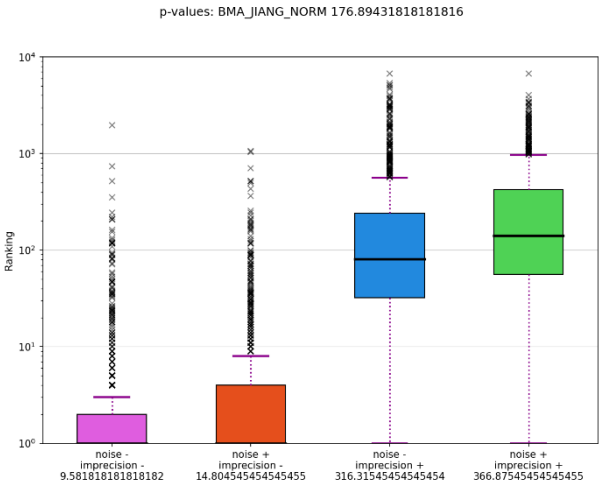
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	2977.5
Noise + Imprecision -	0	2461.5
Noise - Imprecision +	0	2873.5
Noise + Imprecision +	0	2405.0

Pairwise: Jiang and Conrath 1997 Normalized

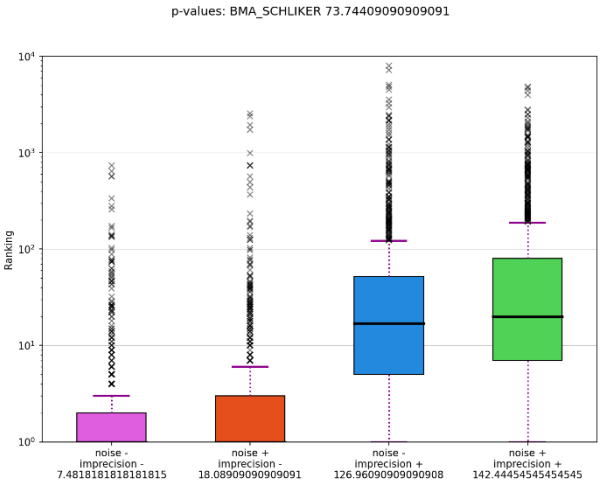
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	745	1.0
Noise + Imprecision -	604	1.0
Noise - Imprecision +	9	81.0
Noise + Imprecision +	3	140.5

Pairwise: Schlicker SimRel 2006

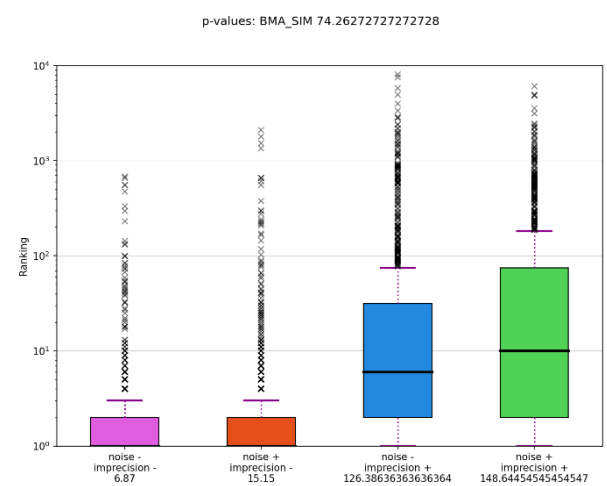
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	754	1.0
Noise + Imprecision -	655	1.0
Noise - Imprecision +	101	17.0
Noise + Imprecision +	81	20.0

Pairwise: SimIC 2010

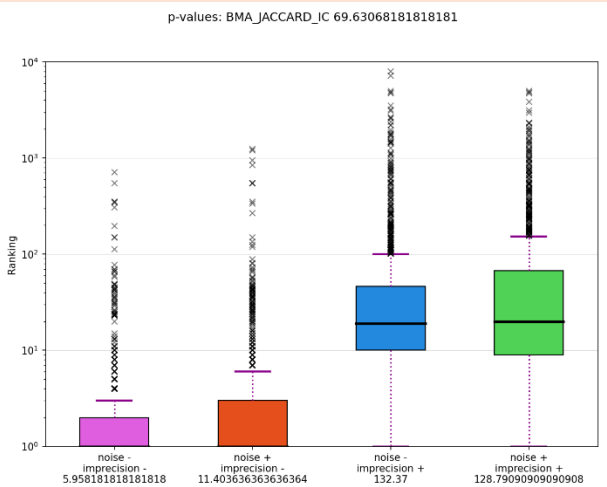
Indirect measure: Best Match Average (BMA)



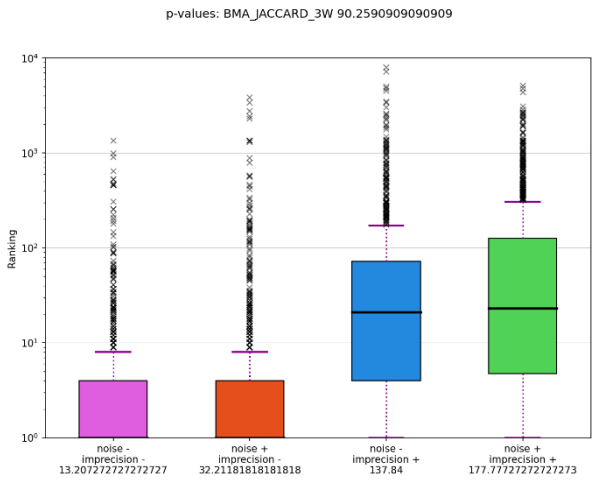
	Nº of 1st ranks	Medians
Noise - Imprecision -	803	1.0
Noise + Imprecision -	692	1.0
Noise - Imprecision +	223	6.0
Noise + Imprecision +	190	10.0

Pairwise: Jaccard IC

Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	806	1.0
Noise + Imprecision -	605	1.0
Noise - Imprecision +	41	19.0
Noise + Imprecision +	22	20.0

Pairwise: Jaccard 3W IC **Indirect measure:** Best Match Average (BMA)			

	Nº of 1st ranks	Medians
Noise - Imprecision -	645	1.0
Noise + Imprecision -	649	1.0
Noise - Imprecision +	129	21.0
Noise + Imprecision +	138	23.0

The best method is SimIC 2010 with p-values, which outperforms Köhler et al. 2009 using similarity scores when noise and imprecision are added.

	Nº of 1st ranks	Medians
Noise - Imprecision -	984	1
Noise + Imprecision -	891	1
Noise - Imprecision +	673	1
Noise + Imprecision +	172	14

Table 17. Performance statistics by similarity scores of Köhler et al. 2009 method

	Nº of 1st ranks	Medians
Noise - Imprecision -	803	1.0
Noise + Imprecision -	692	1.0
Noise - Imprecision +	223	6.0
Noise + Imprecision +	190	10.0

Table 18. Performance statistics by p-values of SimIC 2010 method

However, SimIC falls behind by a huge margin compared with Köhler et al. 2009 in "Noise-Imprecision+". Moreover, the bias is higher.

In conclusion, the only good candidate to substitute Köhler et al. 2009 method could be SimIC. However, the little increase in performance with noise and imprecision is not compensated by the loss of accuracy and higher bias in classifying diseases in the rest of the combinations of those terms. That means that as the symptom set is cleaned from noise and imprecision, the increase in accuracy won't be as high as with the original Köhler et al. 2009.

Conclusions

The disease classification system described in Köhler et al. 2009 was correctly implemented as stated in chapter 4.2 because the performance in the local implementation and the Phenomizer were almost identical. This arises the question of the reliability of the results shown in the original paper (Fig. 9). The median presented in that figure for the p-value method is 1, even with noise and imprecision. The discrepancy could mean that the patients generated had not a high degree of imprecision or noise, or maybe the results were a bit optimistic.

For computing the p-values, simulations using 5000 random sets of symptoms were used instead of 100000 for a matter of time and computational resources, but only a bit of precision was lost because of this change, as explained in chapter 3.3.

With the local ranking system correctly working, many simulations were completed comparing the performance using simply similarity scores, with the p-values method as described by Köhler et al. 2009. It was found that similarity scores performed better than p-values, however, with p-values, in principle, you can see if the results returned by the simulation are trustworthy or not. Chapter 4.3, is shown that the reliability of those p-values for this task is not very high, so the best way was found to be the similarity score method.

Looking for more ways of calculating similarity scores, a lot of new methods were tested using the *SML* library (9). The SimIC method with p-values showed a high performance with noise and imprecision, but, as a drawback, this is not compensated by the fast loss of accuracy and higher bias in classifying diseases in the rest of the combinations of noise and imprecision when “cleaning up” the symptoms set.

Overall, the best method found is a simple similarity score using the equation described in Köhler et al. 2009 (Equation 1).

In future projects, a new similarity score method based on machine learning techniques specifically for this task could be developed.

Bibliografia

(1) Köhler, Sebastian, Marcel H. Schulz, Peter Krawitz, Sebastian Bauer, Sandra Dölken, Claus E. Ott, Christine Mundlos, Denise Horn, Stefan Mundlos, and Peter N. Robinson. 2009. "Clinical Diagnostics In Human Genetics With Semantic Similarity Searches In Ontologies". *The American Journal Of Human Genetics* 85 (4): 457-464. Doi:10.1016/j.ajhg.2009.09.003.

(2) "Human Phenotype Ontology". 2022. *Hpo.Jax.Org*. https://hpo.jax.org/app/download/ontology.

(3) "Human Phenotype Ontology". 2022. *Hpo.Jax.Org*. https://hpo.jax.org/app/download/annotation.

(4) Harispe, Sébastien. 2022. "Semantic Measures Library And Toolkit". *Semantic-Measures-Library.Org*. https://web.archive.org/web/20180827004653/http://www.semantic-measures-library.org/sml/index.php?q=sml-semantic-measures.

(5) Köhler, Sebastian. 2009. *Clinical Diagnostics In Human Genetics With Semantic Similarity Searches In Ontologies*. Ebook. American Journal of Human Genetics, Volume 85 Supplemental Data.

(6) "Human Phenotype Ontology | NCBO Bioportal". 2022. *Bioportal.Bioontology.Org*. https://bioportal.bioontology.org/ontologies/HP.

(7) Köhler, Sebastian. 2009. "Disease Annotations Database". https://compbio.charite.de/65henomi/job/hpo.annotations/.

(8) Harispe, Sébastien. 2022. "Slib SML". *Maven Repository*. https://mvnrepository.com/artifact/com.github.sharispe/slib-sml/0.9.1.

(9) Harispe, Sébastien. 2022. "Semantic Measures Library And Toolkit". *Web.Archive.Org*. https://web.archive.org/web/20171024203921/http://www.semantic-measures-library.org/sml/.

(10) "HPO Annotation File Formats — HPO Annotation Q/C 1 Documentation". 2022. *Hpo-Annotation-Qc.Readthedocs.Io*. https://hpo-annotation-qc.readthedocs.io/en/latest/annotationFormat.html#phenotype-hpoa-format.

(11) "Linear Interpolation – Wikipedia". 2022. *En.Wikipedia.Org*. https://en.wikipedia.org/wiki/Linear_interpolation.

(12) Köhler, Sebastian. 2022. *Supplemental Data Of Clinical Diagnostics In Human Genetics With Semantic Similarity Searches In Ontologies*. Ebook. American Journal of Human Genetics, Volume 85.

(13) "Statistical Tests: Multiple Comparisons". 2022. *Sia.Webpopix.Org*. http://sia.webpopix.org/statisticalTests2.html.

(14) Köhler, Sebastian. 2022. “The Phenomizer – Clinical Diagnostics With Similarity Searches In Ontologies”. *Compbio.Charite.De*. https://compbio.charite.de/phenomizer/.

(15) Harispe, Sébastien. 2022. "Semantic Measures Library And Toolkit". *Web.Archive.Org*. https://web.archive.org/web/20180827004653/http://www.semantic-measures-library.org/sml/index.php?q=sml-semantic-measures.

Annex

Scores comparisons between 100000 and 5000 samples

a. Adrenoleukodystrophy (31 symptoms)

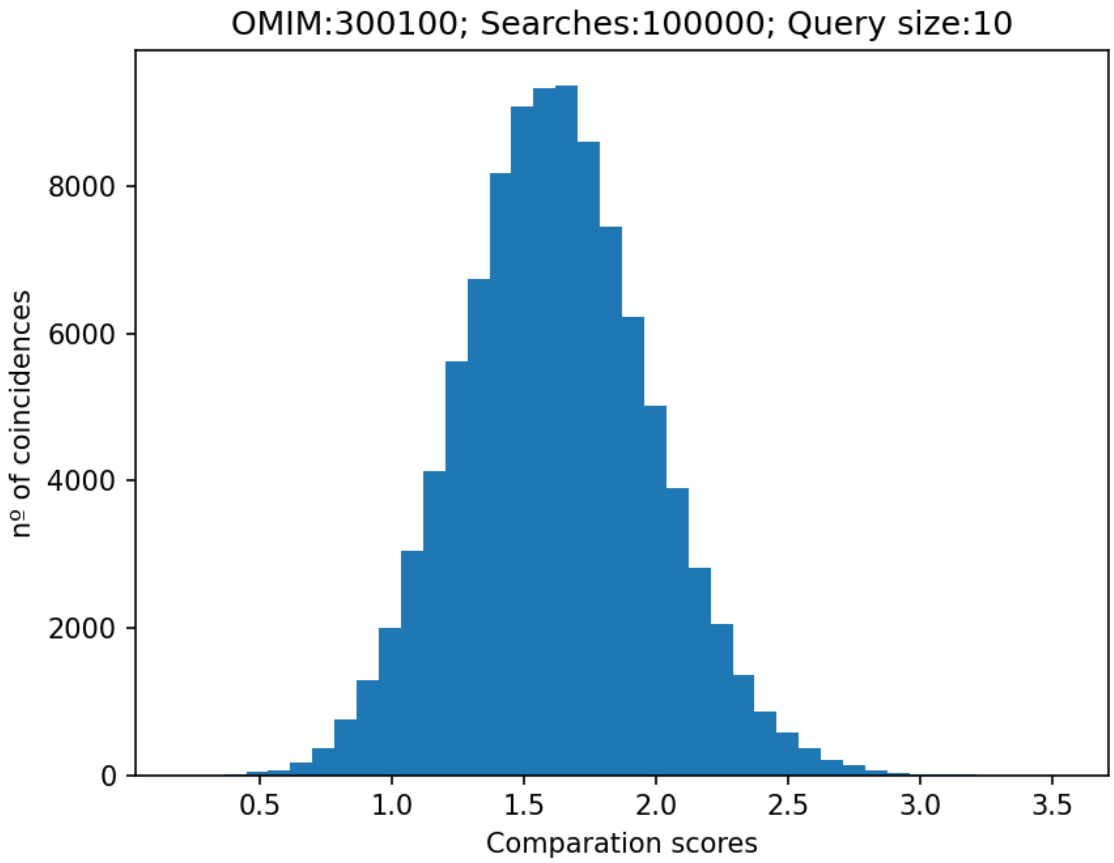


Figure 24. Histogram of the Similarity scores for disease OMIM:300100 (Adrenoleukodystrophy) with 100000 searches and query size of 10

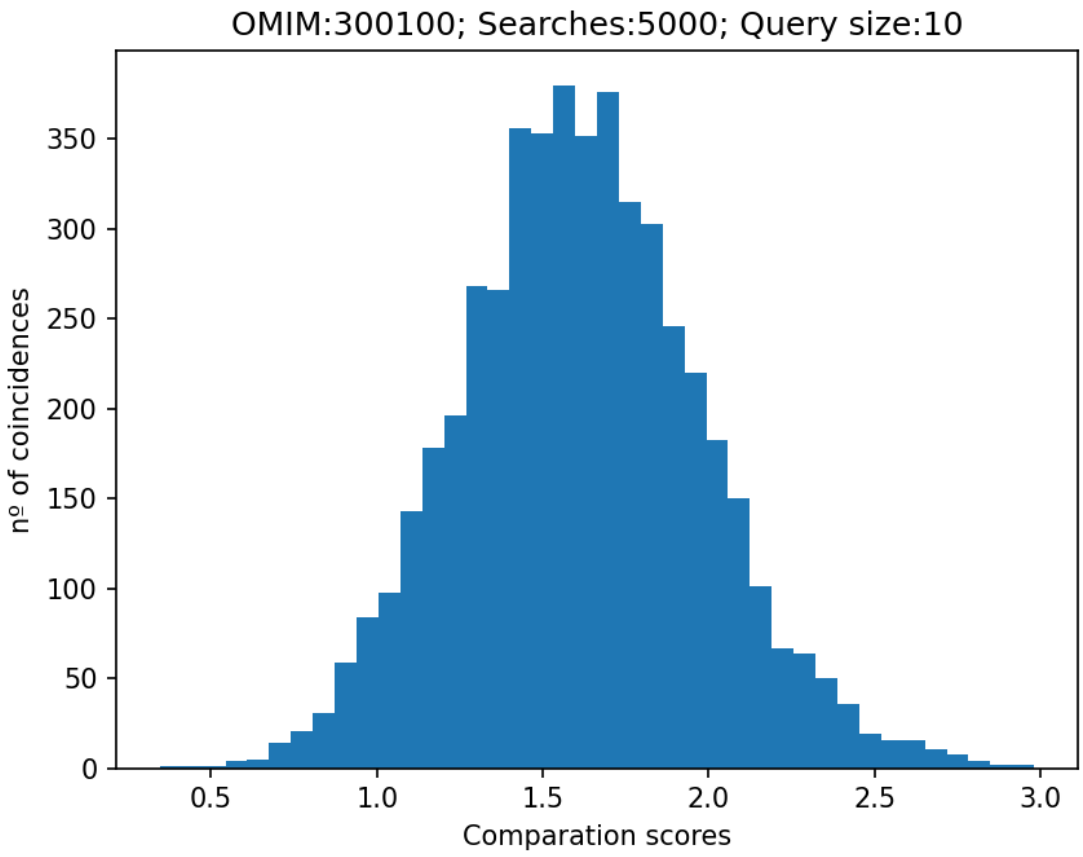


Figure 25 Histogram of the similarity scores for disease OMIM:300100 (Adrenoleukodystrophy) with 5000 searches and query size of 10

p-values pattern	***Scores 100000***	***Scores 5000***
1	0.19683938	0.34558508
0.9	1.166516	1.1509366
0.8	1.3175824	1.3057898
0.7	1.4282119	1.425651
0.6	1.5223223	1.5167466
0.5	1.6123456	1.6051809
0.4	1.7012159	1.6975774
0.3	1.799328	1.7945406

0.2	1.9190576	1.9135044
0.1	2.0878434	2.076246
0.09	2.1111438	2.1003182
0.08	2.1371796	2.1211715
0.07	2.1651263	2.1542656
0.06	2.1968553	2.1877959
0.05	2.230914	2.2266886
0.04	2.2724805	2.2840247
0.03	2.3271668	2.340757
0.02	2.3975418	2.4162076
0.01	2.5145826	2.5516486
0.005	2.618476	2.6567533
0.001	2.8296351	2.812456
0	3.5460958	2.9810739

Table 19. Scores comparison between 100000 and 5000 samples for Adrenoleukodystrophy

b. Megabladder (9 symptoms)

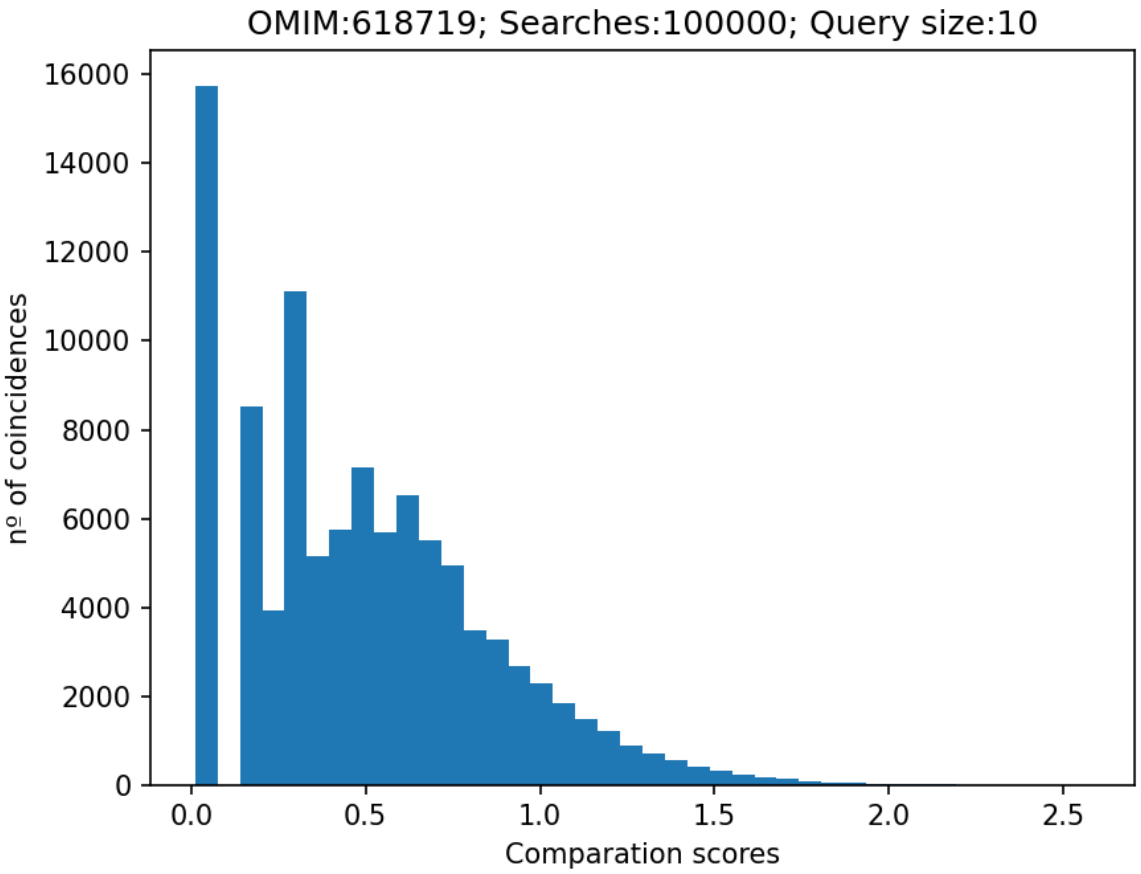


Figure 26. Histogram of the similarity scores for disease OMIM:618719 (Megabladder) with 100000 searches and query size of 10

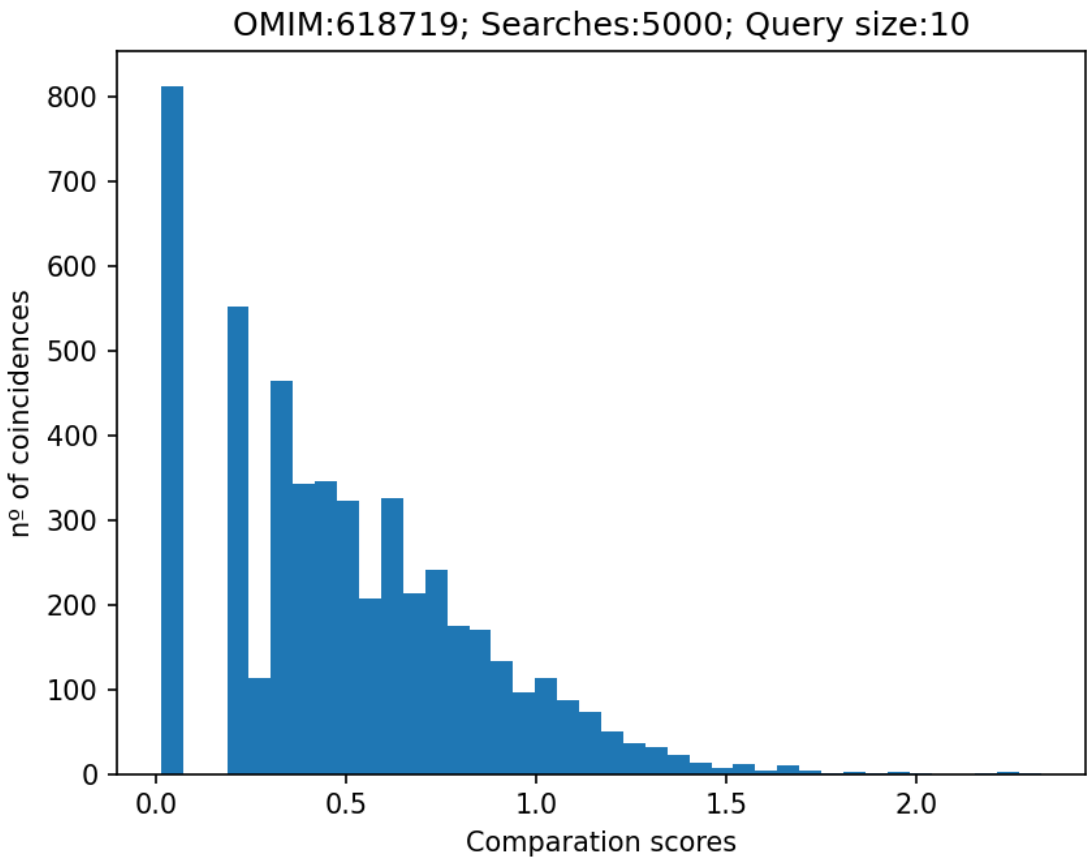


Figure 27 Histogram of the similarity scores for disease OMIM:618719 (Megabladder) with 5000 searches and query size of 10

p-values pattern	***Scores 100000***	***Scores 5000***
1	0.011425085	0.011425085
0.9	0.011425085	0.011425085
0.8	0.18966815	0.18966815
0.7	0.30250037	0.30250037
0.6	0.3665668	0.3665668
0.5	0.4554392	0.45476174
0.4	0.54661614	0.54661614
0.3	0.65764207	0.6530567

0.2	0.7803006	0.77650833
0.1	0.9886342	0.98359066
0.09	1.0160875	1.0102637
0.08	1.0461065	1.0385344
0.07	1.0778913	1.0666656
0.06	1.1171958	1.1020137
0.05	1.1613585	1.1319216
0.04	1.2132193	1.1776403
0.03	1.278928	1.2355825
0.02	1.3702499	1.3195453
0.01	1.5114512	1.4707377
0.005	1.6381904	1.6581869
0.001	1.9240036	1.9818192
0	2.575947	2.3311734

Table 20. Scores comparison between 100000 and 5000 samples for megabladder

Due to the lower number of symptoms (9), is more likely to obtain scores closer to 0, because is harder, probabilistically speaking, to find with the query symptoms that pertain to the disease. Also, could mean that some symptoms that describe this disease are too general, or too specific to one branch of the HPO.

Code Index

In this section is provided an index of the files used for some specific calculations in this thesis. The files are adjoined in a zip file together with this report.

- Transform format database from *hpoa* to *tsv*: **HPOA_to_TSV.py**
- P-values calculation: **Monte_Carlo.java**
- P-values plots: **Monte_Carlo_Plots.java**
- Simulated patients' generation: **Simulate_Patients.java**
- Phenomizer results scrapping: **Test_Phenomizer.java**
- Phenomizer diseases database scraping: **Get_HPOA_From_Phenomizer.java**
- Boxplots and statistics: **BoxPlotResults.py**

Limited imprecision method

Limit → 3

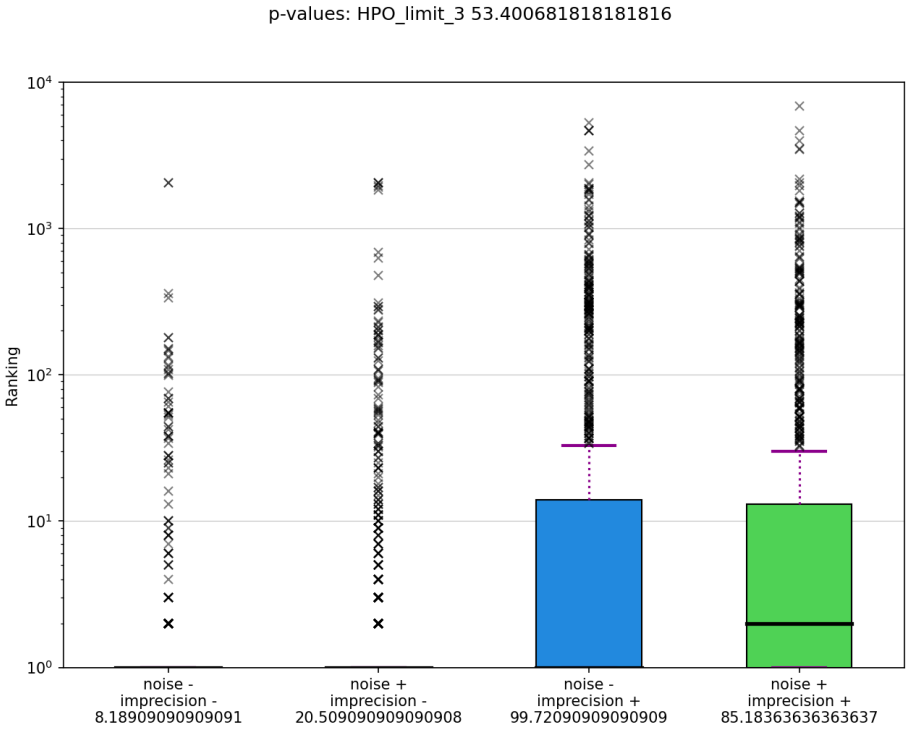


Figure 28. P-values method with imprecision limit set to 3

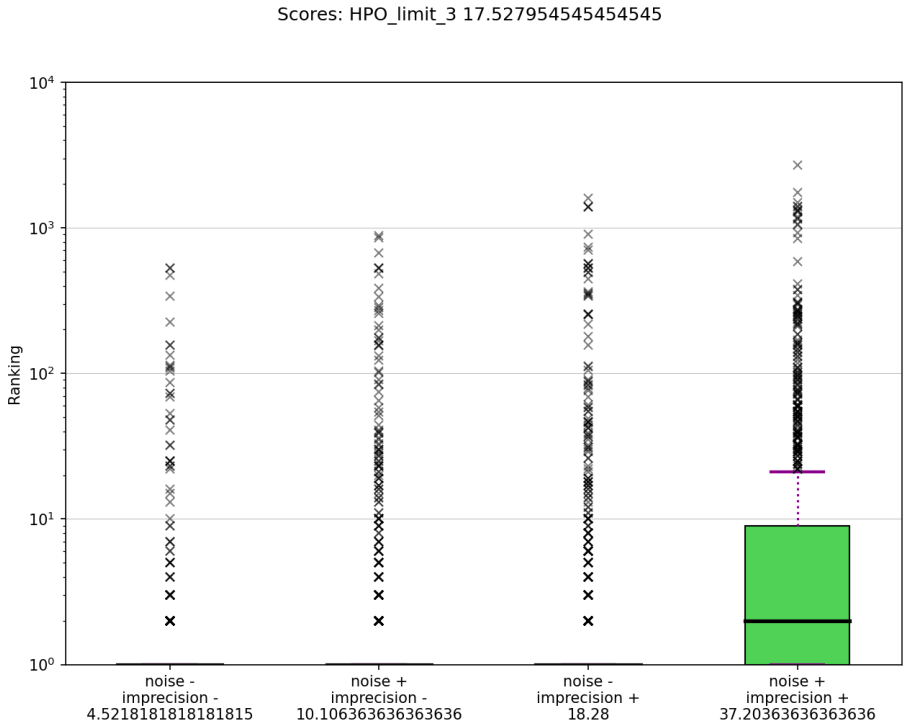


Figure 29. Similarity scores method with imprecision limit set to 3

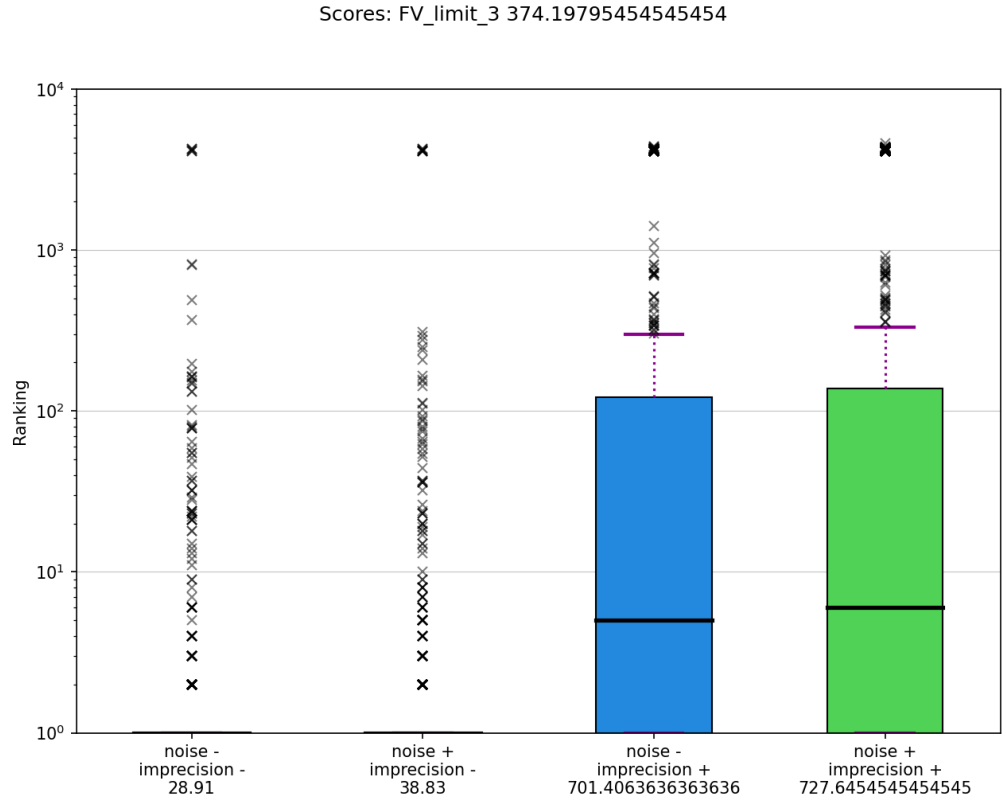


Figure 30. FV method with imprecision limit set to 3

Limit → 4

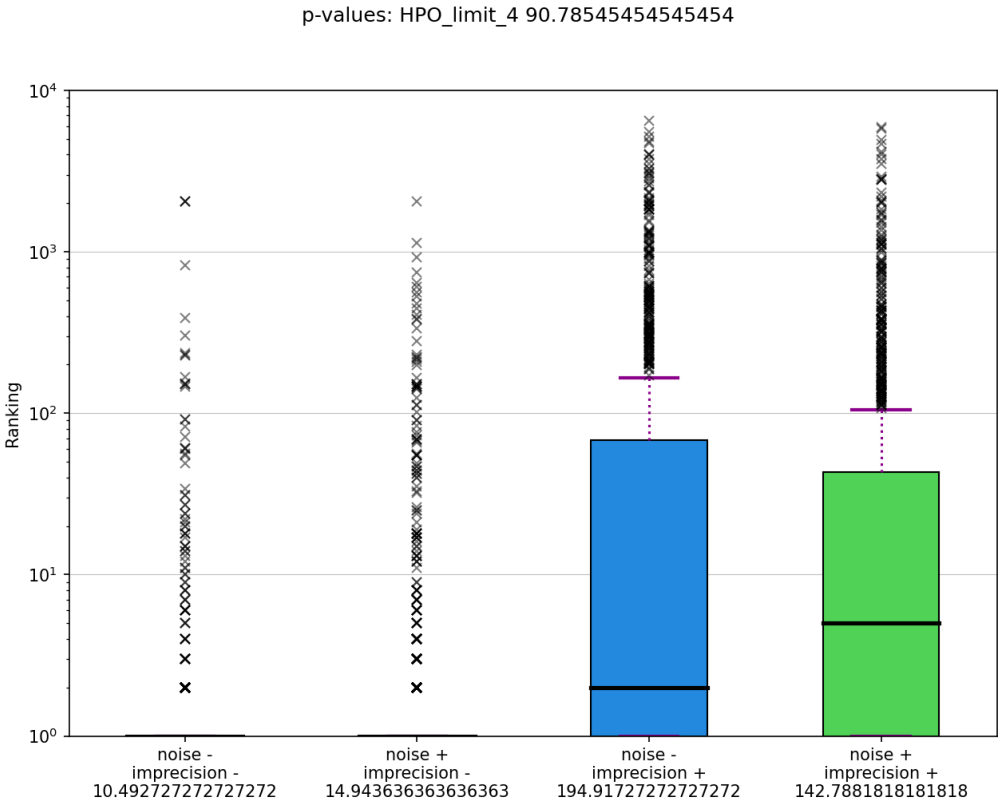


Figure 31. P-values method with imprecision limit set to 4

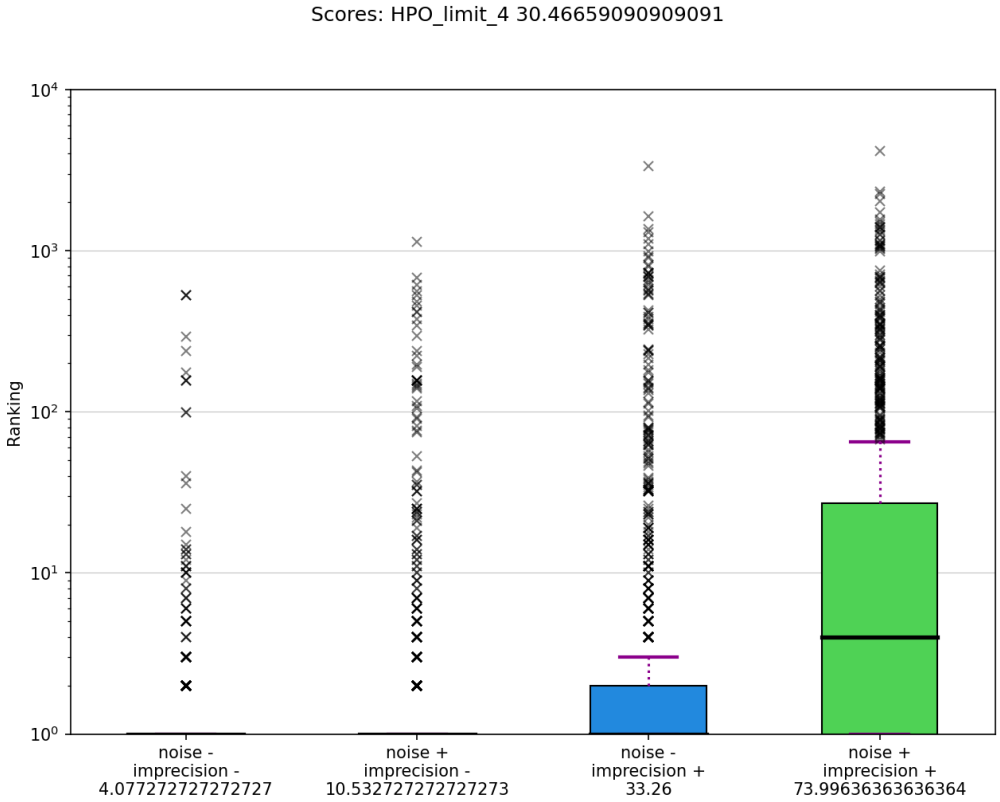
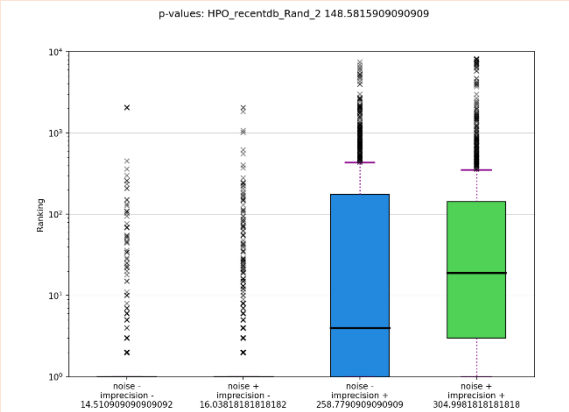


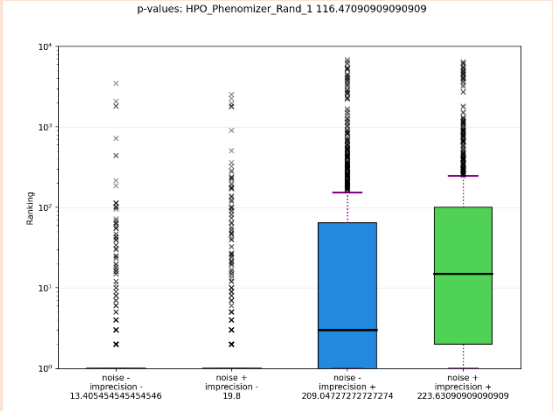
Figure 32. Similarity scores method with imprecision limit set to 4

Performance comparison between old and latest databases

Methods	Using the latest databases (25/05/2022)	Using databases close in time with the paper

P-values method

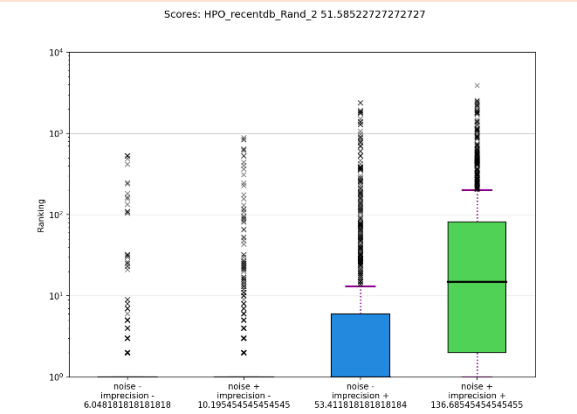


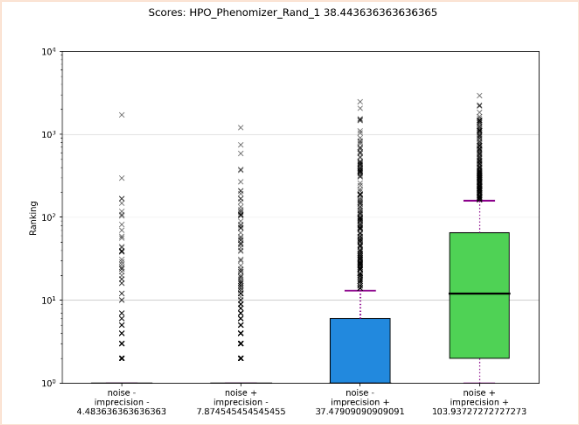


	Nº of 1st ranks	Medians
Noise – Imprecision -	986	1
Noise + Imprecision -	871	1
Noise – Imprecision +	399	4
Noise + Imprecision +	195	19

	Nº of 1st ranks	Medians
Noise – Imprecision -	946	1
Noise + Imprecision -	867	1
Noise – Imprecision +	465	3
Noise + Imprecision +	195	15

Similarity Scores





	Nº of 1st ranks	Medians
Noise – Imprecision -	990	1
Noise + Imprecision -	887	1
Noise – Imprecision +	647	1

	Nº of 1st ranks	Medians
Noise – Imprecision -	985	1
Noise + Imprecision -	878	1
Noise – Imprecision +	658	1

	Noise + Imprecision +	211	15	**Noise + Imprecision +**	213	12

Table 21. Comparison using new and old databases

The similarity scores method performs better than the p-value method in terms of better median and a greater number of 1st ranks. Also, the old database seems to perform a bit better than the latest versions.

Phenomizer vs local simulation

Latest databases comparison

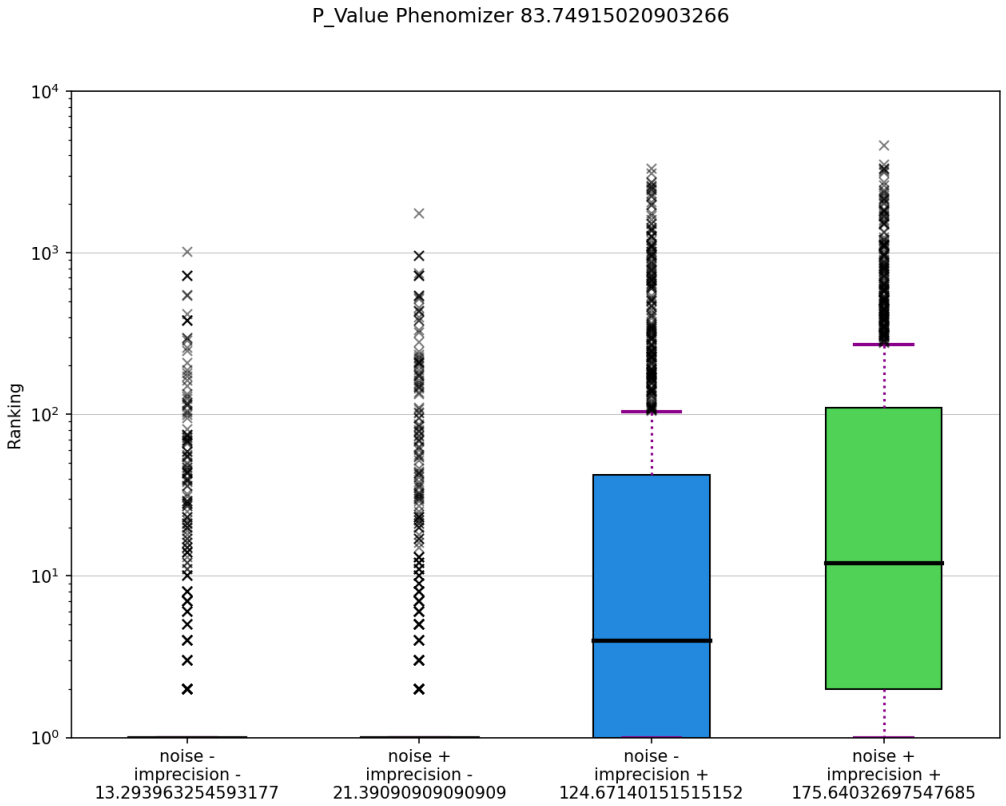


Figure 33. Phenomizer performance using the P-value method, the patients Set 2, and the latest databases

	Nº of 1st ranks	**Medians**

Noise – Imprecision -	942	1
Noise + Imprecision -	854	1
Noise – Imprecision +	372	4
Noise + Imprecision +	241	12

Table 22. Figure 31 statistics

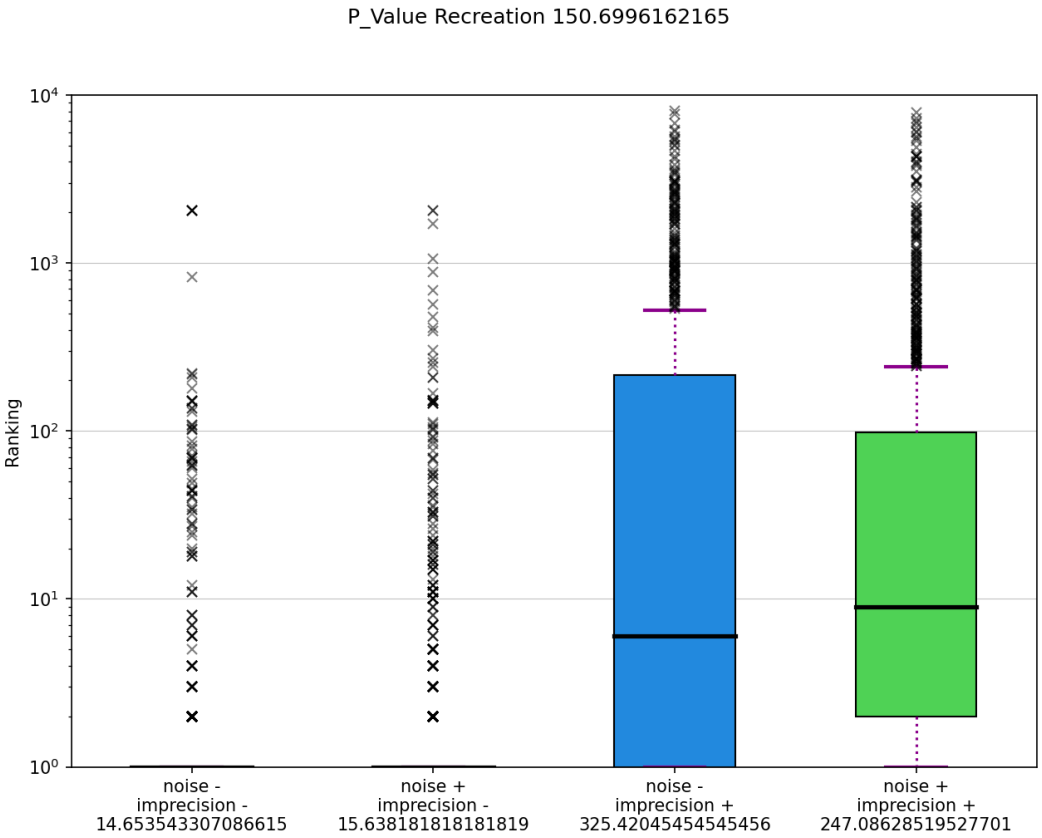


Figure 34. Local simulation performance using the P-value method, the patients Set 2, and the latest databases

	Nº of 1st ranks	**Medians**
Noise – Imprecision -	995	1
Noise + Imprecision -	886	1
Noise – Imprecision +	387	6

Noise + Imprecision +	264	9

Table 23. Figure 32 statistics

The local simulation seems to perform better than the Phenomizer but the median in "noise-imprecision+" is higher due to the high bias.

Older databases comparison

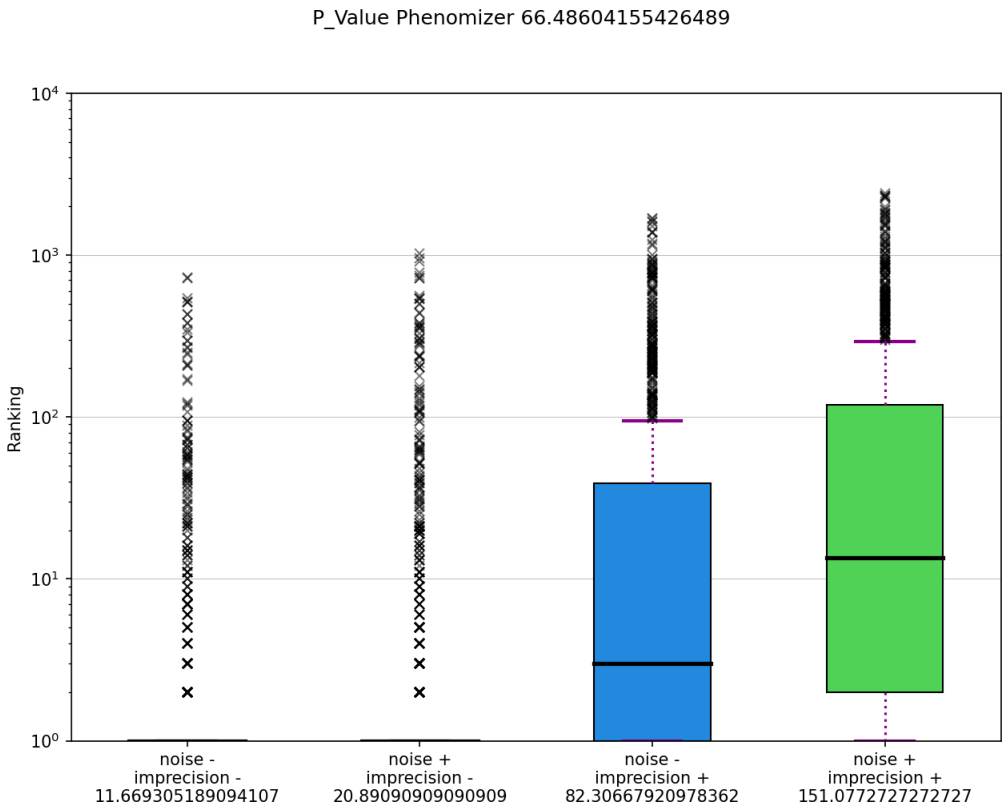


Figure 35. Phenomizer performance using the P-value method, the patients Set 3, and older databases

	Nº of 1st ranks	**Medians**
Noise – Imprecision -	931	1
Noise + Imprecision -	837	1
Noise – Imprecision +	410	3

Noise + **Imprecision +**	231	13.5

Table 24. Figure 33 statistics

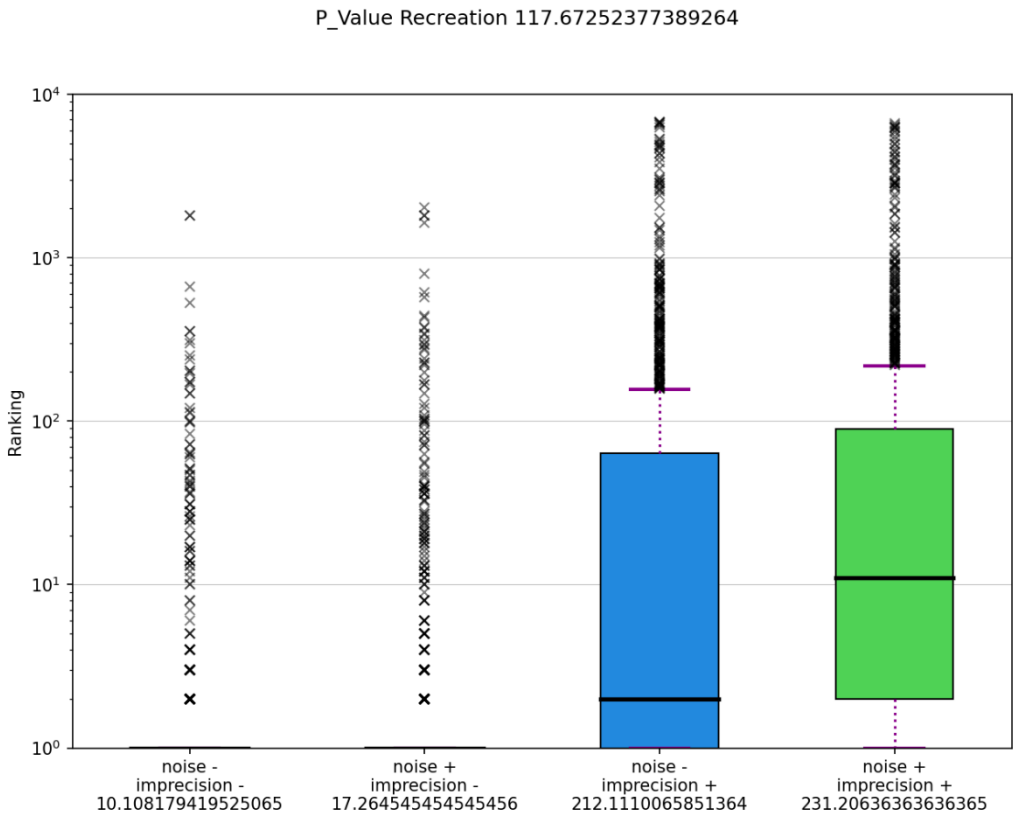


Figure 36. Local simulation performance using the P-value method, the patients Set 3, and older databases

	Nº of 1st ranks	**Medians**
Noise – **Imprecision -**	972	1
Noise + **Imprecision -**	862	1
Noise – **Imprecision +**	451	2
Noise + **Imprecision +**	202	11

Table 25. Figure 34 statistics

In this case, the bias in the Phenomizer in "noise+ imprecision+" is higher than in the local simulation but the number of 1st ranks is also higher. On the other hand, the performance of "noise- imprecision+" is better in the local simulation.

Comparing the old and newest databases, the older performs a bit worse in terms of the number of 1st ranks, but the bias is also lower.

P-value Truth Tables

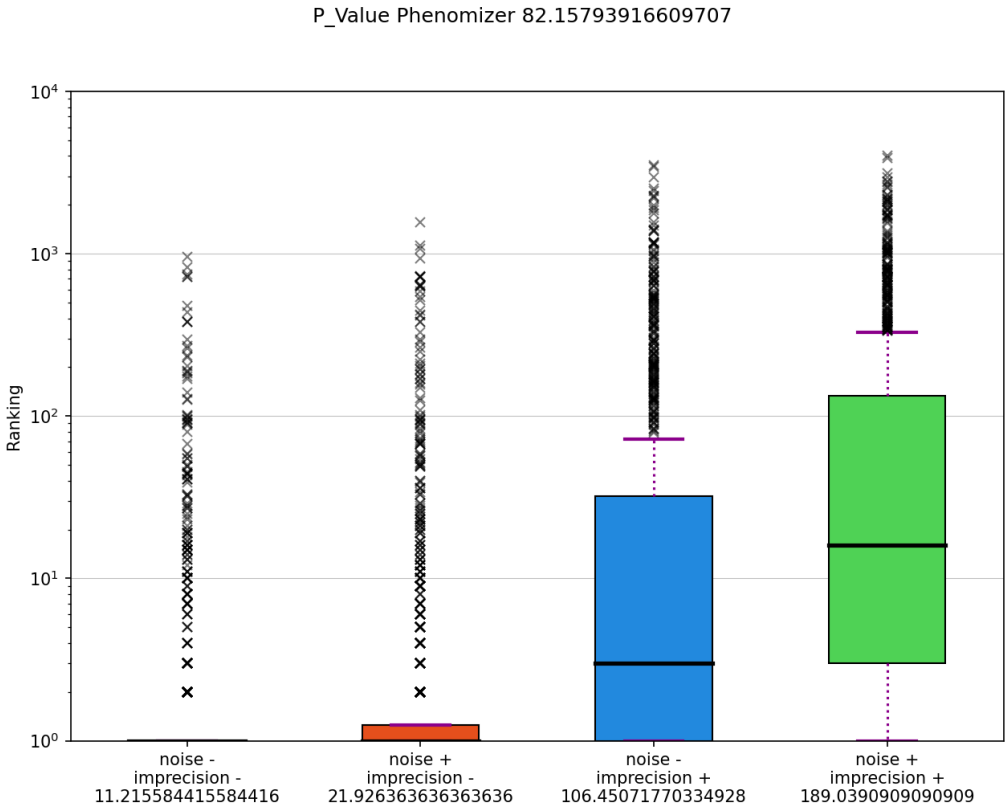


Figure 37. Phenomizer performance using the P-value method, and older databases

Phenomizer 0 (latest databases) (Rank < 1)

	False Negative	True Positive	False Positive	True Negative
Noise – Imprecision -	30	926	110	79
Noise + Imprecision -	80	745	152	123
Noise – Imprecision +	314	84	81	566
Noise + Imprecision +	163	25	43	869
Total	587	1780	396	1637

Table 26. Truth table of Figure 35

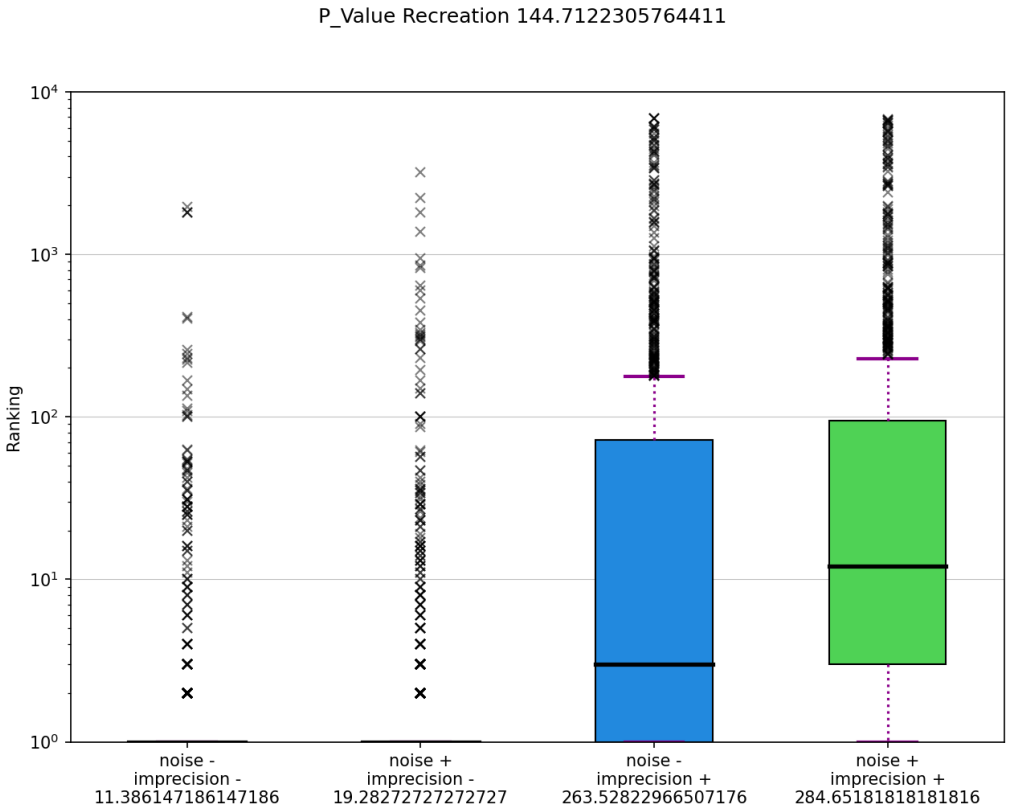


Figure 38. Local simulation performance using the P-value method, and older databases

Local simulation 0 (latest databases) (Rank < 1)

	False Negative	True Positive	False Positive	True Negative
Noise – Imprecision -	6	978	149	22
Noise + Imprecision -	135	725	201	39
Noise – Imprecision +	331	80	296	338
Noise + Imprecision +	155	39	151	755
Total	627	1822	397	1154

Table 27. Truth table of Figure 36

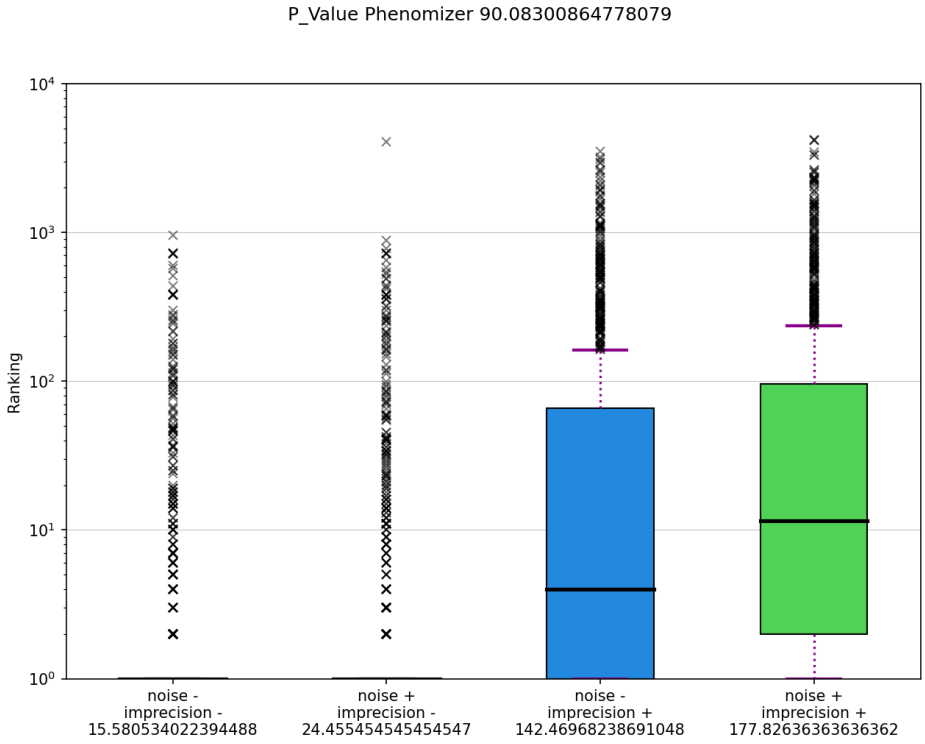


Figure 39. Phenomizer performance using the P-value method, the patient Set 0, and the latest databases

Phenomizer 1 (latest databases) (Rank < 1)

	False Negative	True Positive	False Positive	True Negative
Noise - Imprecision -	16	923	143	79
Noise + Imprecision -	70	794	143	93
Noise - Imprecision +	259	76	94	610
Noise + Imprecision +	181	33	73	813
Total	526	1826	453	1595

Table 28. Truth table of Figure 22

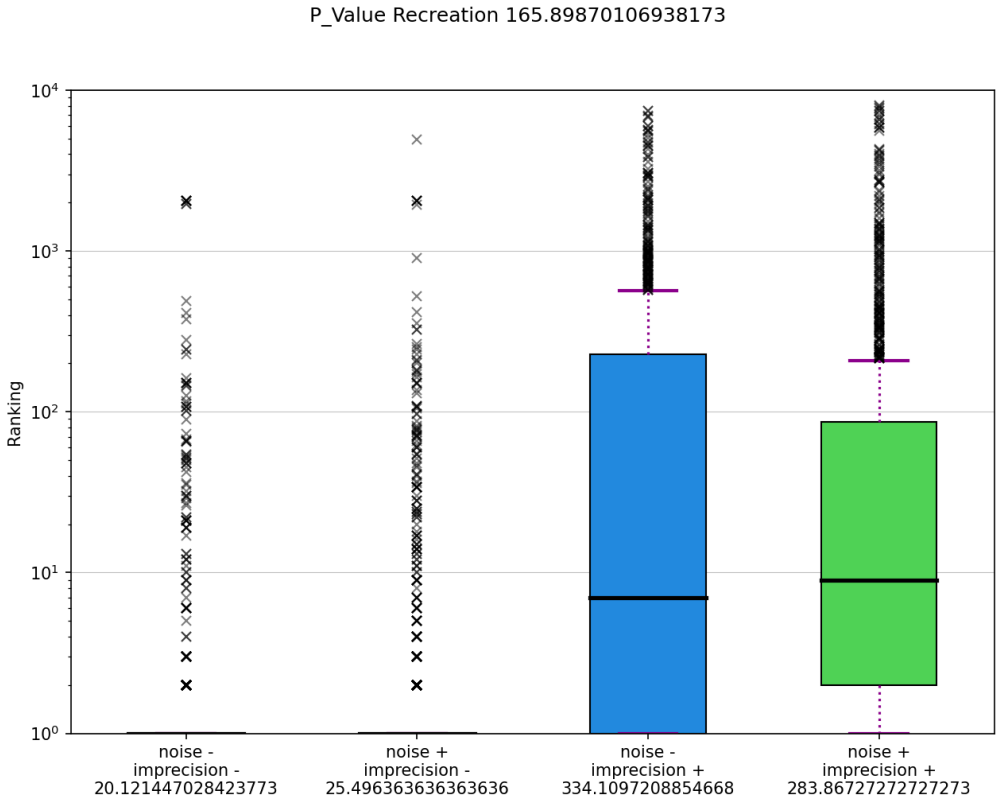


Figure 40. Local simulation performance using the P-value method, the patient Set 0, and the latest databases

Local simulation 1 (latest databases) (Rank < 1)

	False Negative	True Positive	False Positive	True Negative
Noise – Imprecision -	1	1009	145	6
Noise + Imprecision -	21	859	202	18
Noise – Imprecision +	239	112	395	293
Noise + Imprecision +	196	53	242	609
Total	457	2033	984	926

Table 29. Truth table of Figure 23

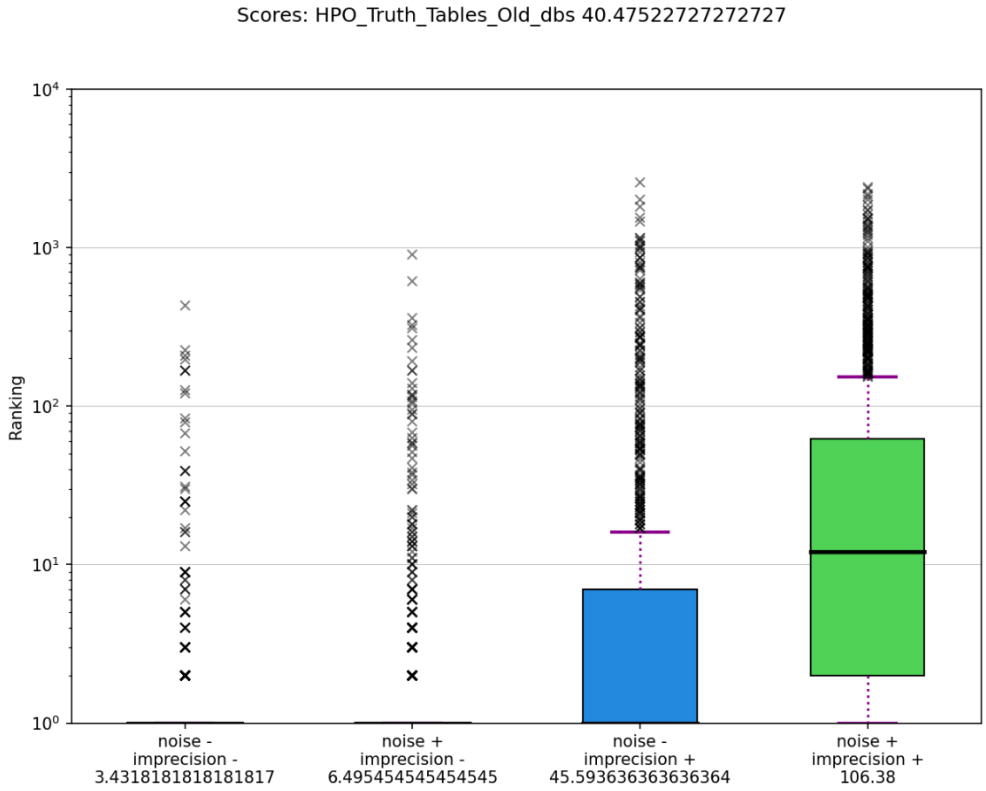


Figure 41. Local simulation using similarity scores with old databases

	Nº of 1st ranks	**Medians**
Noise - **Imprecision -**	980	1
Noise + **Imprecision -**	896	1
Noise - **Imprecision +**	622	1
Noise + **Imprecision +**	212	12

Table 30. Figure 41 boxplot statistics

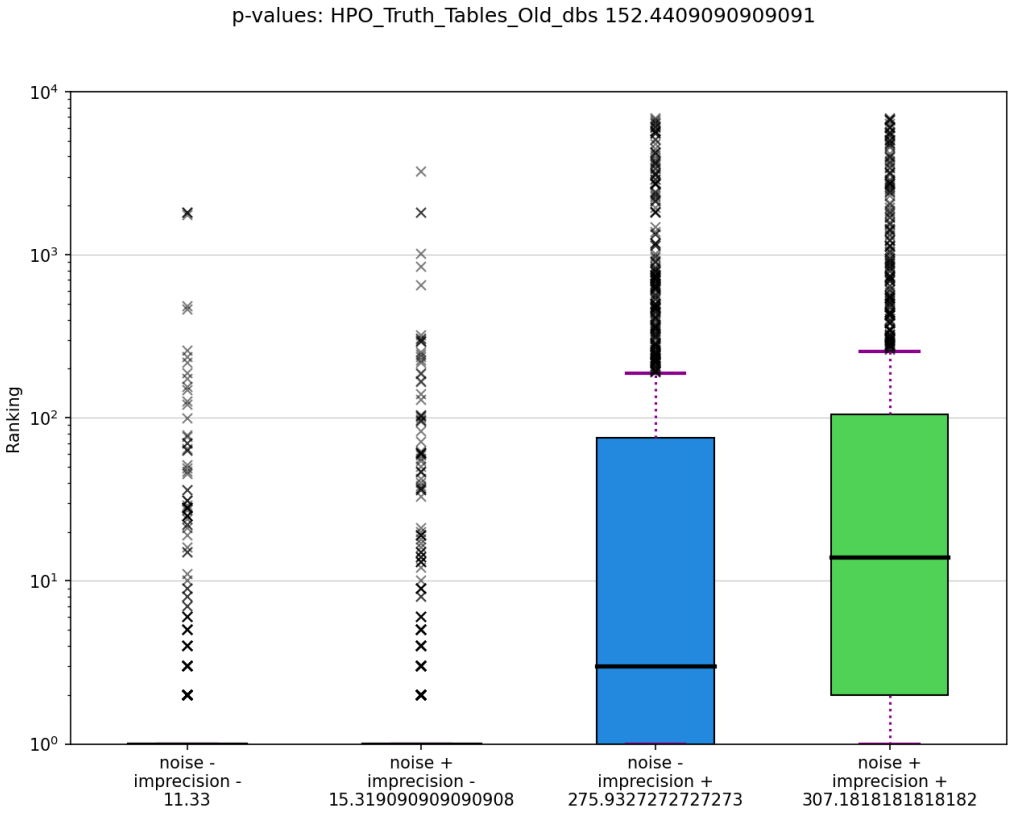


Figure 42. Local simulation using p-values with old databases

	Nº of 1st ranks	Medians
Noise - **Imprecision -**	980	1
Noise + **Imprecision -**	896	1
Noise - **Imprecision +**	622	1
Noise + **Imprecision +**	212	12

Table 31. Figure 42 boxplot statistics

Local simulation (old databases) (Rank <= 1)

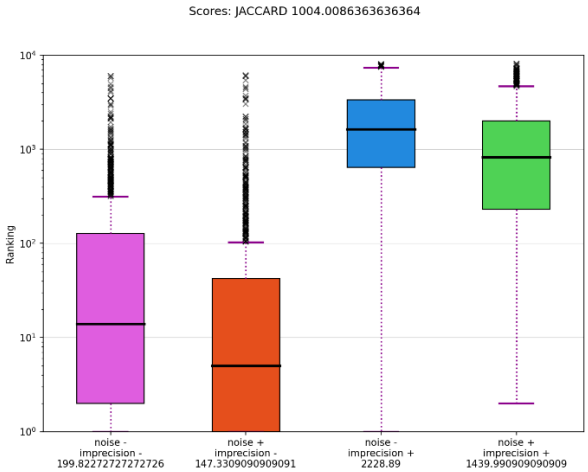
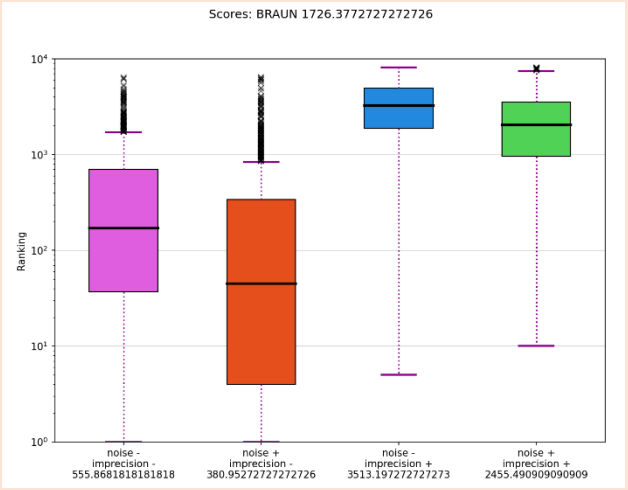
	False Negative	True Positive	False Positive	True Negative
Noise – **Imprecision -**	6	940	143	11
Noise + **Imprecision -**	123	765	181	31
Noise – **Imprecision +**	354	86	305	355

Noise + Imprecision +	185	25	141	749
Total	668	1816	770	1146

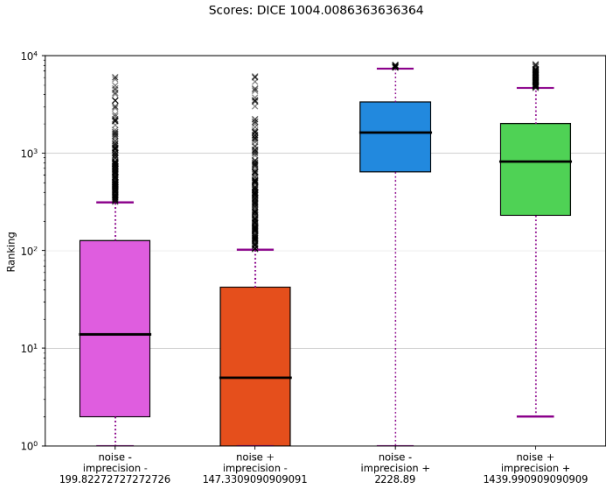
Table 32. Figure 42 truth table

Similarity score methods

Direct Groupwise measures by similarity scores

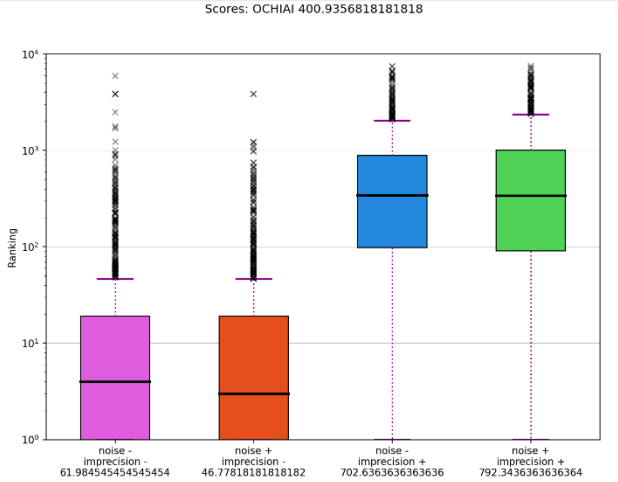
Method	Boxplot	Boxplot performance
Jaccard 1901		**Nº of 1st ranks** / **Medians 1st ranks** **Noise - Imprecision -**: 219 / 14.0 **Noise + Imprecision -**: 364 / 5.0 **Noise - Imprecision +**: 2 / 1635.0 **Noise + Imprecision +**: 0 / 829.0
Braun Blanquet 1932		**Nº of 1st ranks** / **Medians 1st ranks** **Noise - Imprecision -**: 47 / 172.0 **Noise + Imprecision -**: 170 / 45.0 **Noise - Imprecision +**: 0 / 3291.5 **Noise + Imprecision +**: 0 / 2055.5

Dice 1935



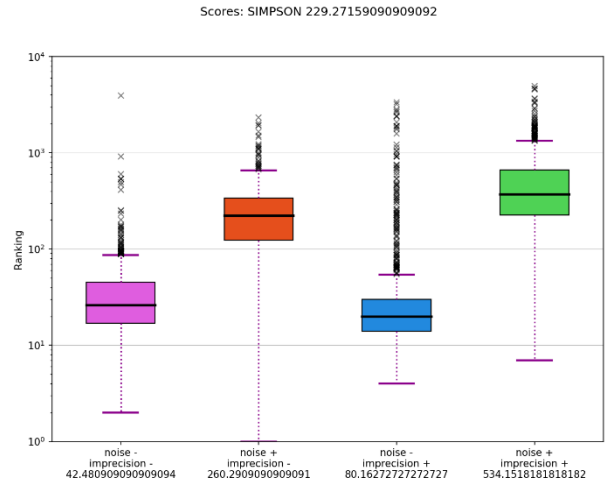
	Nº of 1st ranks	Medians
Noise - Imprecision -	219	14.0
Noise + Imprecision -	364	5.0
Noise - Imprecision +	2	1635.0
Noise + Imprecision +	0	829.0

Ochiai 1957



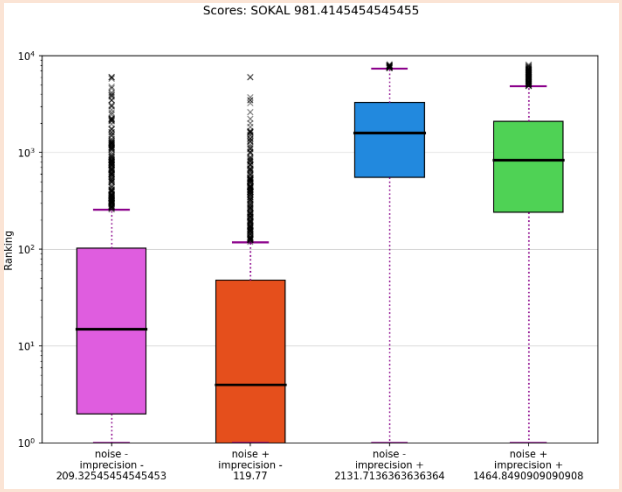
	Nº of 1st ranks	Medians
Noise - Imprecision -	378	4.0
Noise + Imprecision -	427	3.0
Noise - Imprecision +	6	342.5
Noise + Imprecision +	12	341.0

Simpson 1960



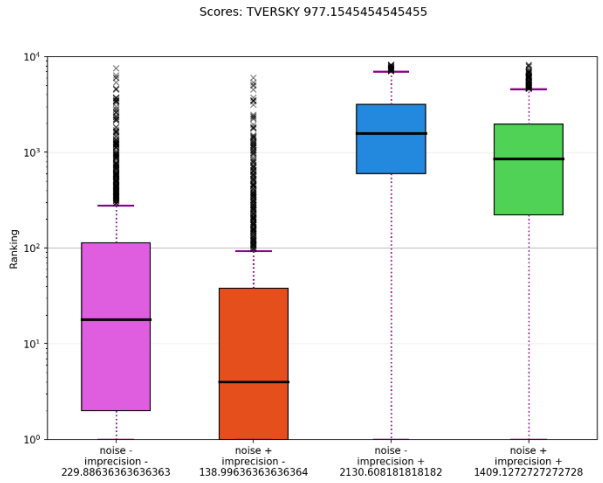
	Nº of 1st ranks	Medians
Noise - Imprecision -	0	26.0
Noise + Imprecision -	1	221.5
Noise - Imprecision +	0	20.0
Noise + Imprecision +	0	372.0

Sokal & Sneath 1963



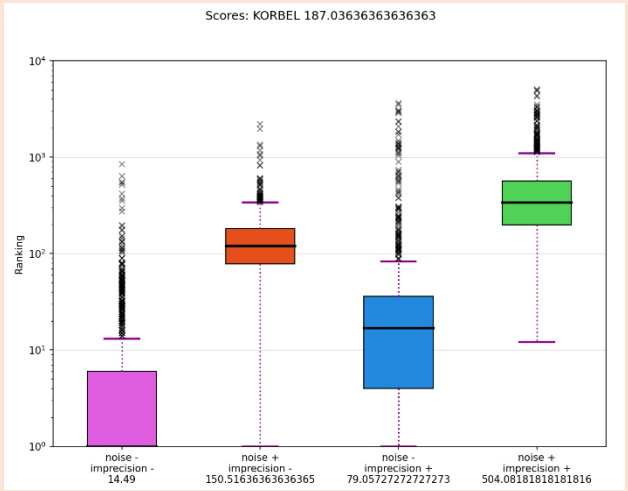
	Nº of 1st ranks	Medians
Noise - Imprecision -	224	15.0
Noise + Imprecision -	389	4.0
Noise - Imprecision +	1	1604.5
Noise + Imprecision +	5	832.5

Tversky 1977 Abstract Model



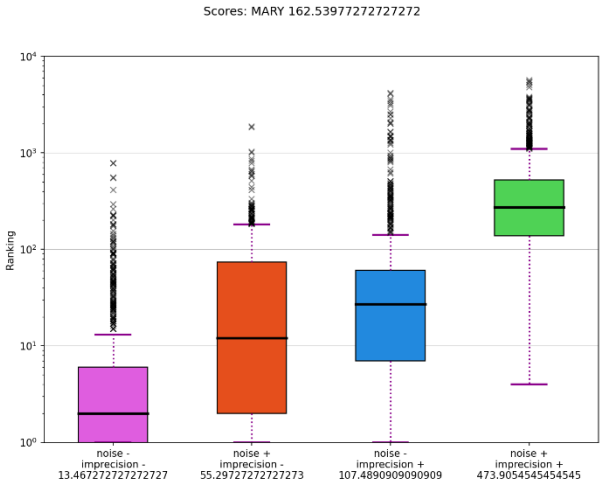
	Nº of 1st ranks	Medians
Noise - Imprecision -	217	18.0
Noise + Imprecision -	389	4.0
Noise - Imprecision +	3	1578.0
Noise + Imprecision +	5	854.5

Korbel 2002



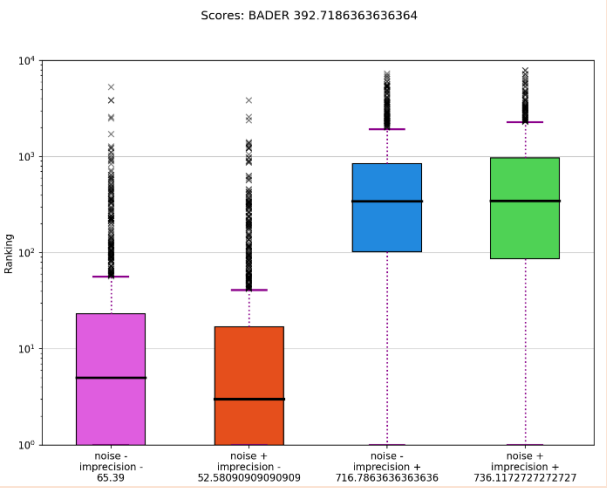
	Nº of 1st ranks	Medians
Noise - Imprecision -	615	1.0
Noise + Imprecision -	11	121.0
Noise - Imprecision +	111	17.0
Noise + Imprecision +	0	340.0

Maryland Bridge 2003



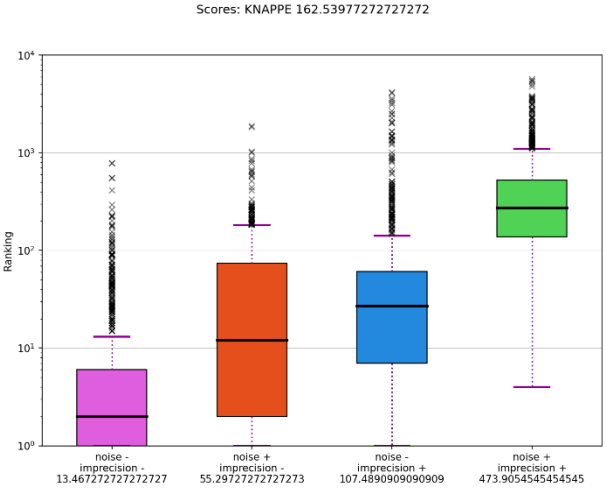
	Nº of 1st ranks	Medians
Noise - Imprecision -	535	2.0
Noise + Imprecision -	268	12.0
Noise - Imprecision +	74	27.0
Noise + Imprecision +	0	272.0

Bader 2003



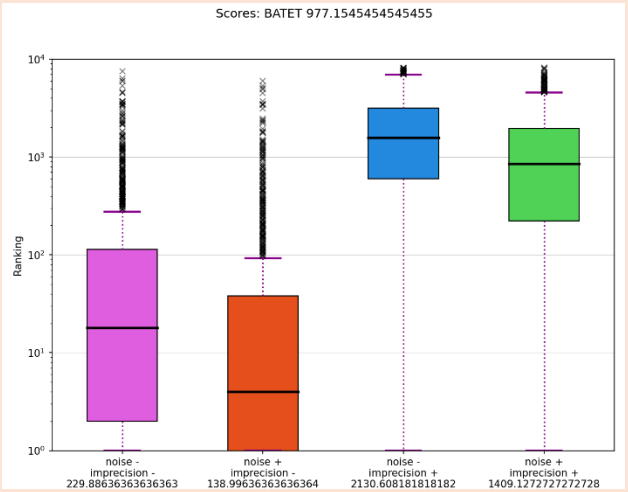
	Nº of 1st ranks	Medians
Noise - Imprecision -	344	5.0
Noise + Imprecision -	435	3.0
Noise - Imprecision +	9	342.5
Noise + Imprecision +	6	348.5

Knappe 2004



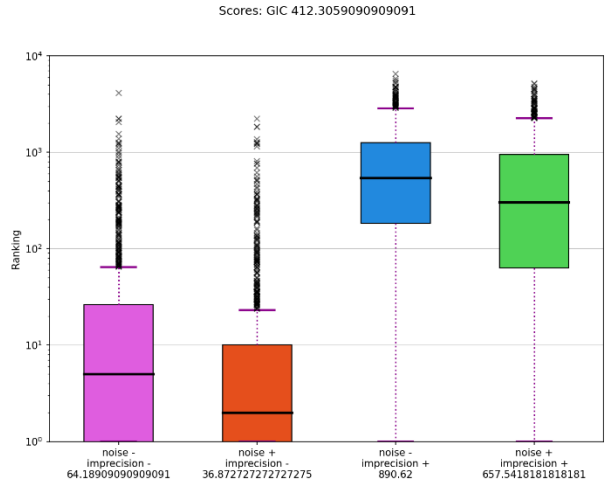
	Nº of 1st ranks	Medians
Noise - Imprecision -	535	2.0
Noise + Imprecision -	268	12.0
Noise - Imprecision +	74	27.0
Noise + Imprecision +	0	272.0

Batet 2010



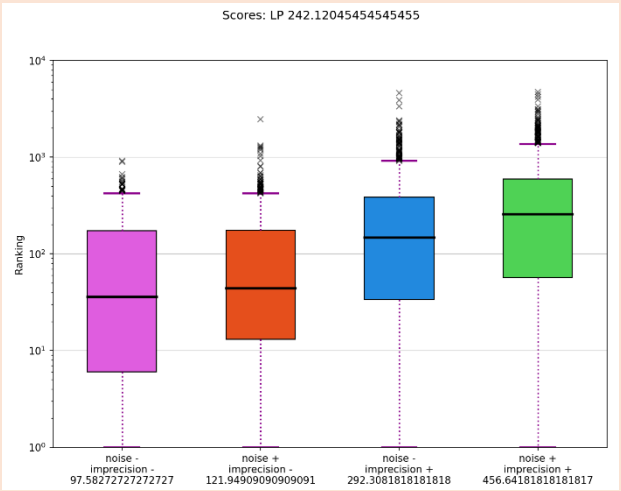
	Nº of 1st ranks	Medians
Noise - Imprecision -	217	18.0
Noise + Imprecision -	389	4.0
Noise - Imprecision +	3	1578.0
Noise + Imprecision +	5	854.5

Sim GIC



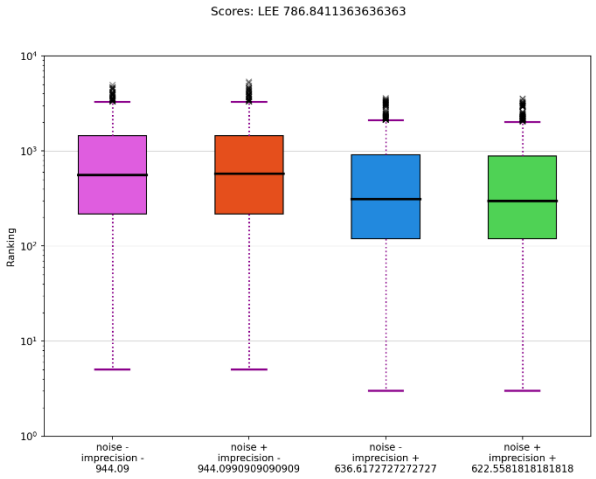
	Nº of 1st ranks	Medians
Noise - Imprecision -	325	5.0
Noise + Imprecision -	526	2.0
Noise - Imprecision +	5	539.5
Noise + Imprecision +	16	304.5

SimLP



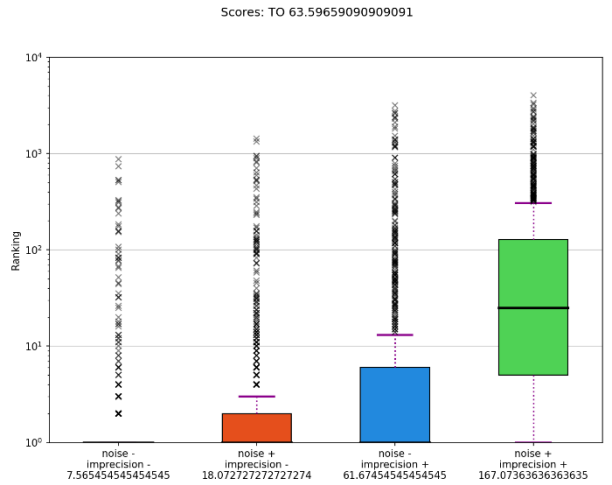
	Nº of 1st ranks	Medians
Noise - Imprecision -	124	36.0
Noise + Imprecision -	92	44.0
Noise - Imprecision +	28	147.0
Noise + Imprecision +	21	257.0

Lee 2004



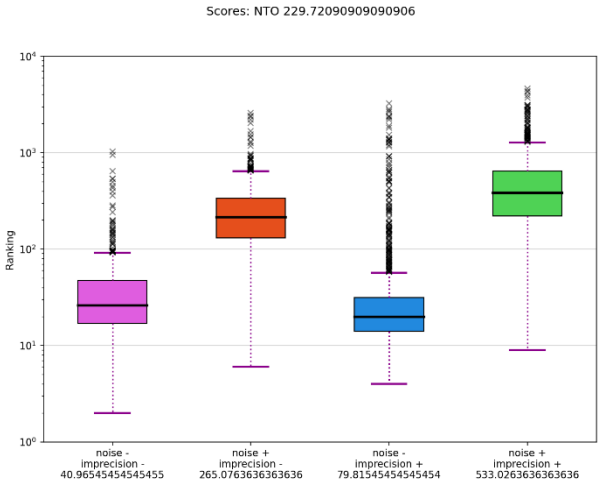
	Nº of 1st ranks	Medians
Noise - Imprecision -	0	561.0
Noise + Imprecision -	0	577.0
Noise - Imprecision +	0	314.0
Noise + Imprecision +	0	299.5

Term Overlap

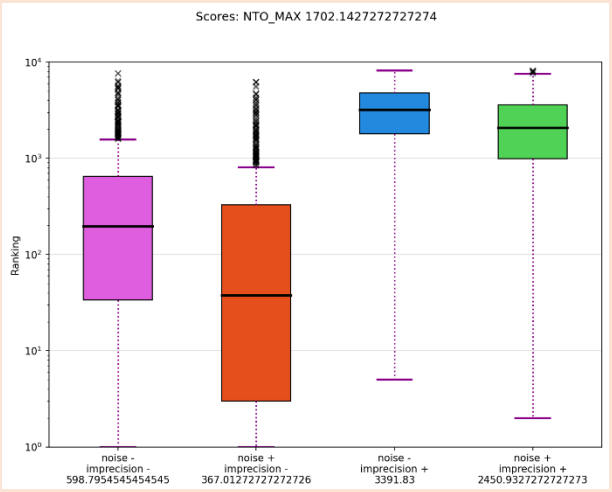


	Nº of 1st ranks	Medians
Noise - Imprecision -	962	1.0
Noise + Imprecision -	748	1.0
Noise - Imprecision +	666	1.0
Noise + Imprecision +	126	25.0

Normalized Term Overlap



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	26.0
Noise + Imprecision -	0	215.0
Noise - Imprecision +	0	20.0
Noise + Imprecision +	0	385.0

			Nº of 1st ranks	Medians
Normalized (Max) Term Overlap		**Noise - Imprecision -**	57	197.0
		Noise + Imprecision -	175	37.5
		Noise - Imprecision +	0	3199.0
		Noise + Imprecision +	0	2064.5

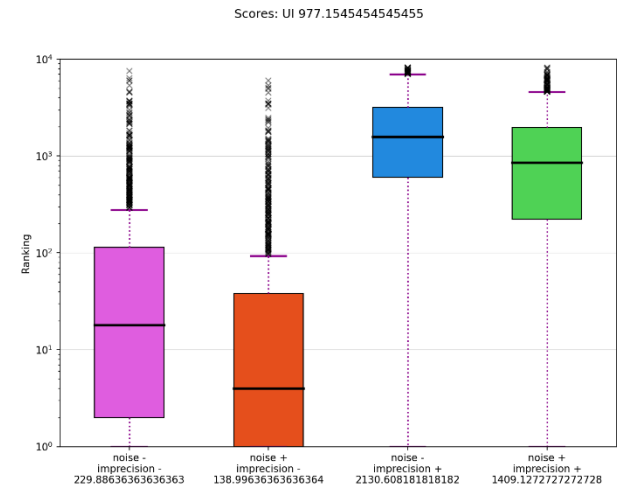
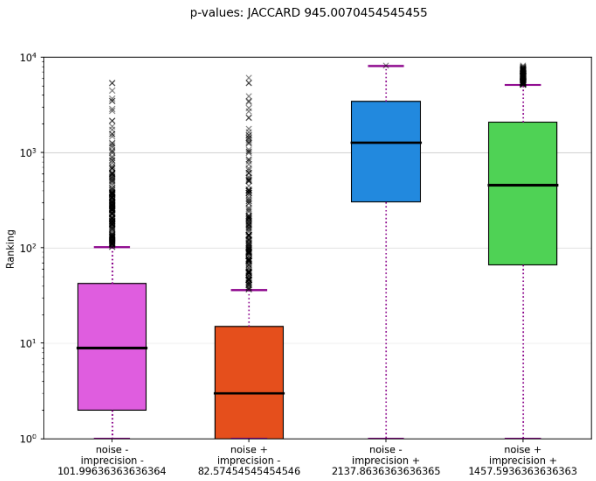
			Nº of 1st ranks	Medians
Sim UI		**Noise - Imprecision -**	217	18.0
		Noise + Imprecision -	389	4.0
		Noise - Imprecision +	3	1578.0
		Noise + Imprecision +	5	854.5

Table 33. Direct Groupwise measures by similarity scores using the latest databases

Direct Groupwise measures by p-values

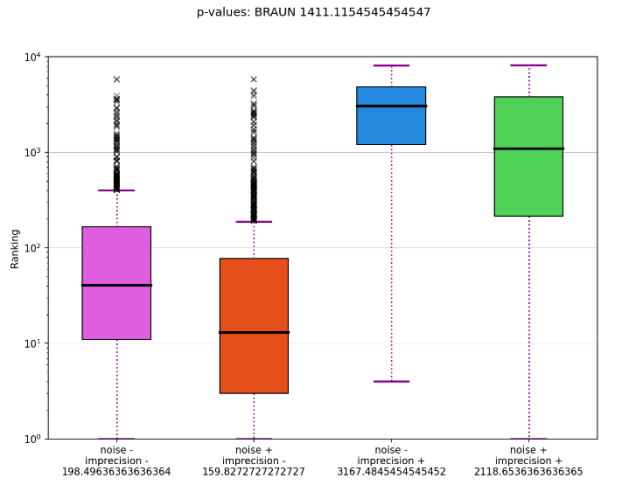
Method	Boxplot	Boxplot performance

Jaccard 1901



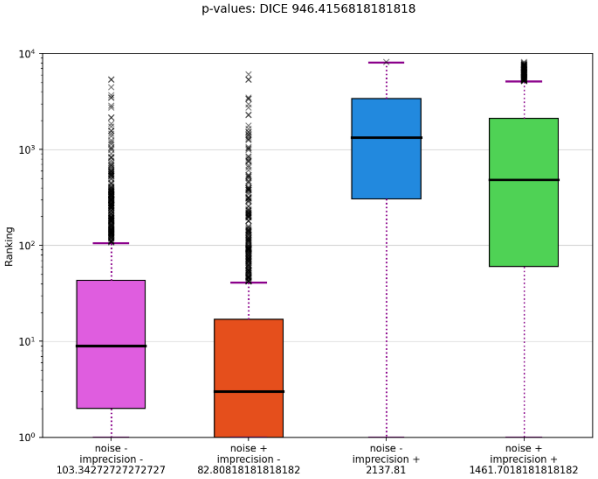
	Nº of 1st ranks	Medians
Noise - Imprecision -	233	9.0
Noise + Imprecision -	391	3.0
Noise - Imprecision +	3	1275.0
Noise + Imprecision +	15	454.0

Braun Blanquet 1932



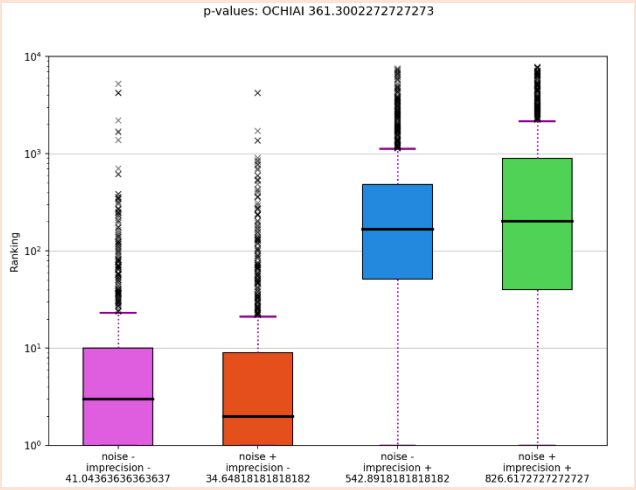
	Nº of 1st ranks	Medians
Noise - Imprecision -	60	41.0
Noise + Imprecision -	188	13.0
Noise - Imprecision +	0	3042.0
Noise + Imprecision +	7	1093.0

Dice 1935



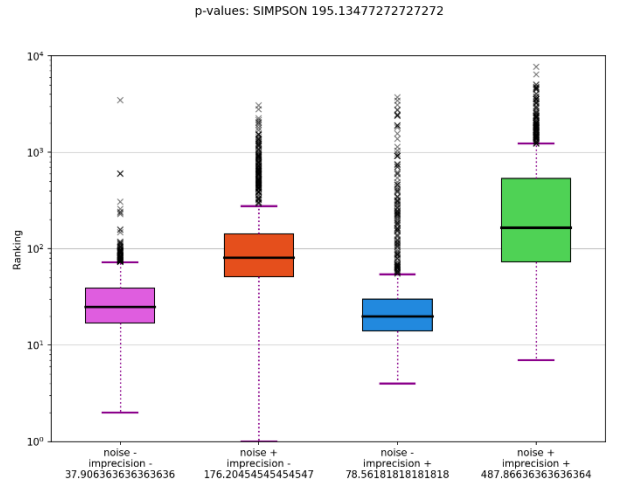
	Nº of 1st ranks	Medians
Noise - Imprecision -	234	9.0
Noise + Imprecision -	391	3.0
Noise - Imprecision +	3	1324.5
Noise + Imprecision +	9	482.0

Ochiai 1957



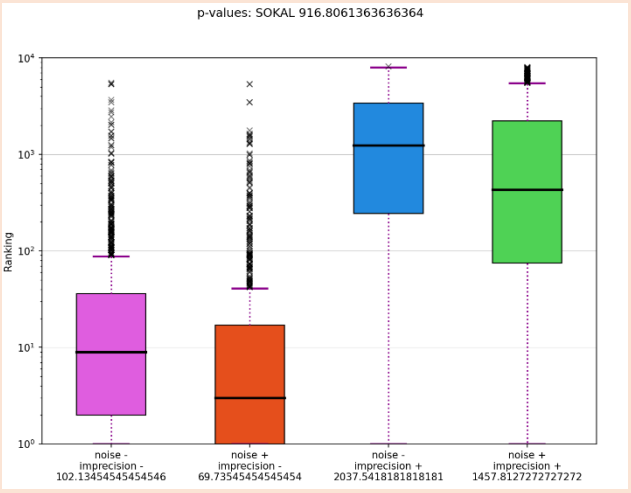
	Nº of 1st ranks	Medians
Noise - Imprecision -	397	3.0
Noise + Imprecision -	462	2.0
Noise - Imprecision +	8	167.5
Noise + Imprecision +	27	204.0

Simpson 1960



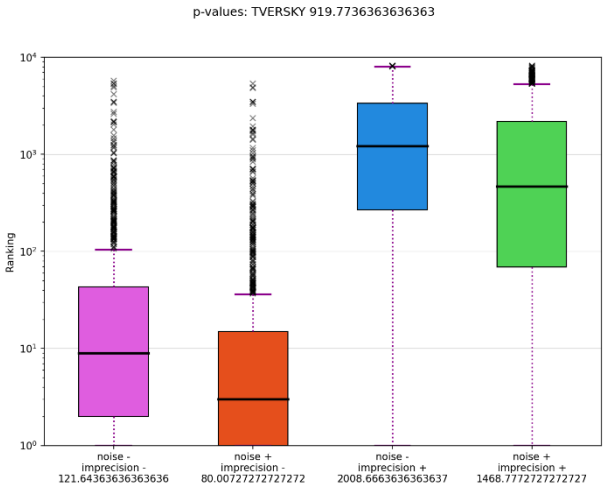
	Nº of 1st ranks	Medians
Noise - Imprecision -	0	25.0
Noise + Imprecision -	1	81.0
Noise - Imprecision +	0	20.0
Noise + Imprecision +	0	166.0

Sokal & Sneath 1963



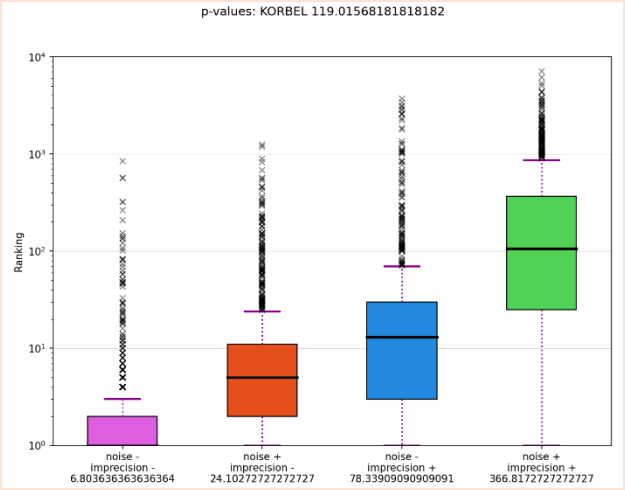
	Nº of 1st ranks	Medians
Noise - Imprecision -	234	9.0
Noise + Imprecision -	410	3.0
Noise - Imprecision +	1	1243.5
Noise + Imprecision +	13	431.5

Tversky 1977 Abstract Model



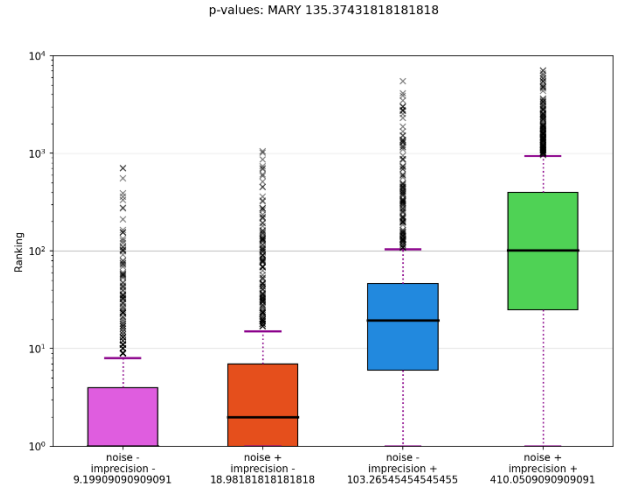
	Nº of 1st ranks	Medians
Noise - Imprecision -	226	9.0
Noise + Imprecision -	412	3.0
Noise - Imprecision +	6	1218.0
Noise + Imprecision +	17	468.0

Korbel 2002



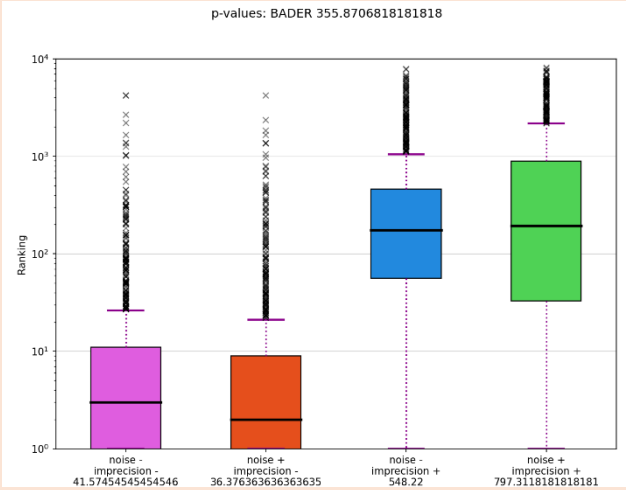
	Nº of 1st ranks	Medians
Noise - Imprecision -	727	1.0
Noise + Imprecision -	139	5.0
Noise - Imprecision +	167	13.0
Noise + Imprecision +	37	106.0

Maryland Bridge 2003



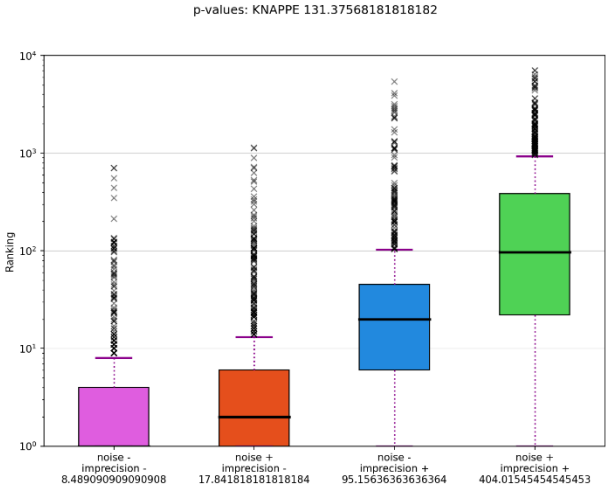
	Nº of 1st ranks	Medians
Noise - Imprecision -	588	1.0
Noise + Imprecision -	396	2.0
Noise - Imprecision +	97	19.5
Noise + Imprecision +	36	101.5

Bader 2003



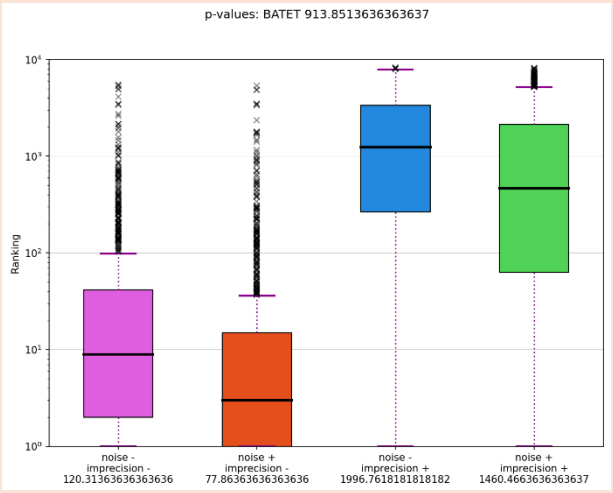
	Nº of 1st ranks	Medians
Noise - Imprecision -	367	3.0
Noise + Imprecision -	477	2.0
Noise - Imprecision +	10	174.5
Noise + Imprecision +	25	194.5

Knappe 2004



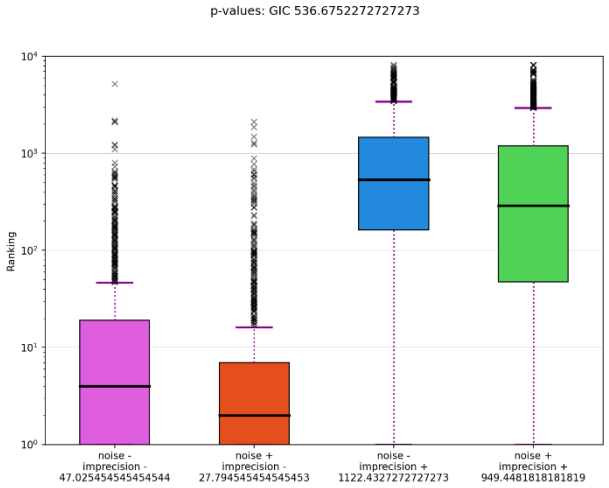
	Nº of 1st ranks	Medians
Noise - Imprecision -	588	1.0
Noise + Imprecision -	406	2.0
Noise - Imprecision +	94	20.0
Noise + Imprecision +	29	96.5

Batet 2010



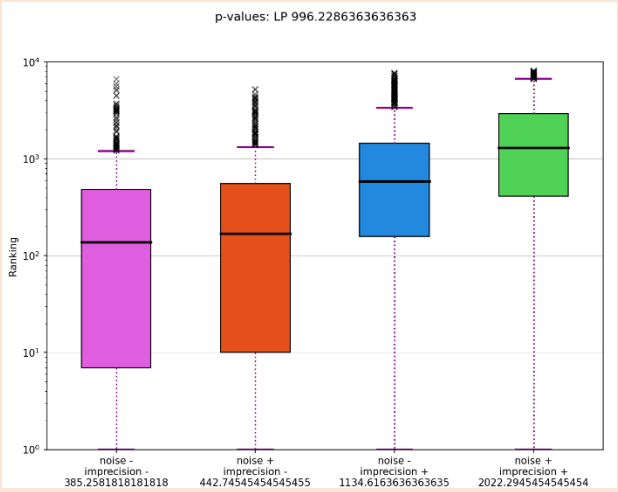
	Nº of 1st ranks	Medians
Noise - Imprecision -	227	9.0
Noise + Imprecision -	419	3.0
Noise - Imprecision +	8	1238.0
Noise + Imprecision +	19	468.0

Sim GIC



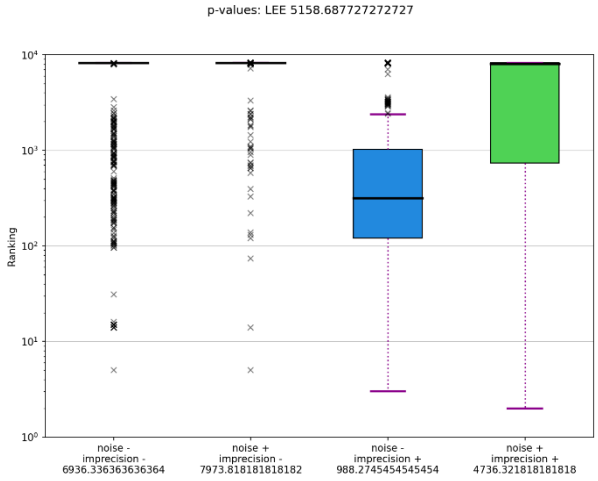
	Nº of 1st ranks	Medians
Noise - Imprecision -	336	4.0
Noise + Imprecision -	547	2.0
Noise - Imprecision +	8	533.5
Noise + Imprecision +	24	288.0

SimLP



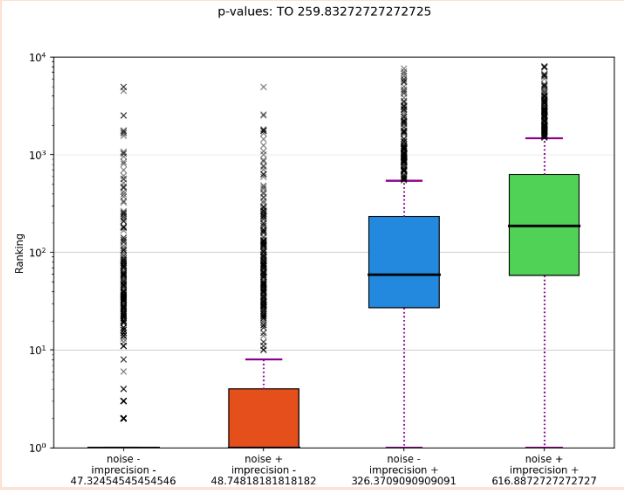
	Nº of 1st ranks	Medians
Noise - Imprecision -	122	137.5
Noise + Imprecision -	99	169.5
Noise - Imprecision +	10	584.5
Noise + Imprecision +	14	1305.5

Lee 2004



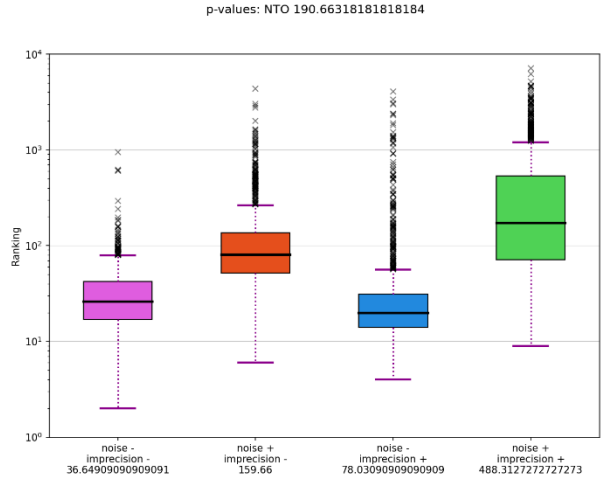
	Nº of 1st ranks	Medians
Noise - Imprecision -	0	8206.0
Noise + Imprecision -	0	8209.0
Noise - Imprecision +	0	317.0
Noise + Imprecision +	0	8052.5

Term Overlap



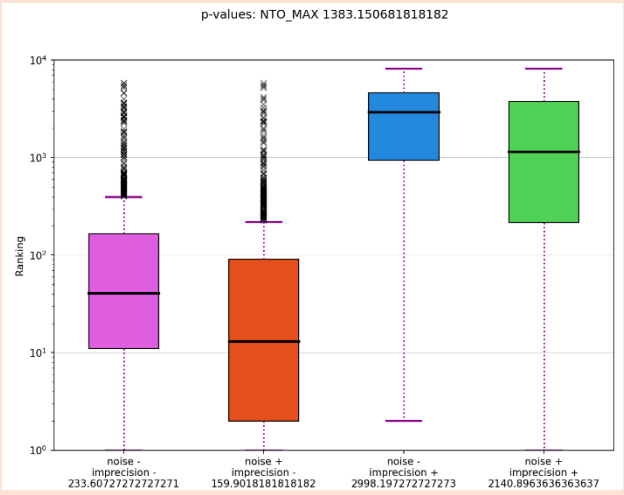
	Nº of 1st ranks	Medians
Noise - Imprecision -	830	1.0
Noise + Imprecision -	723	1.0
Noise - Imprecision +	79	59.5
Noise + Imprecision +	74	187.5

Normalized Term Overlap



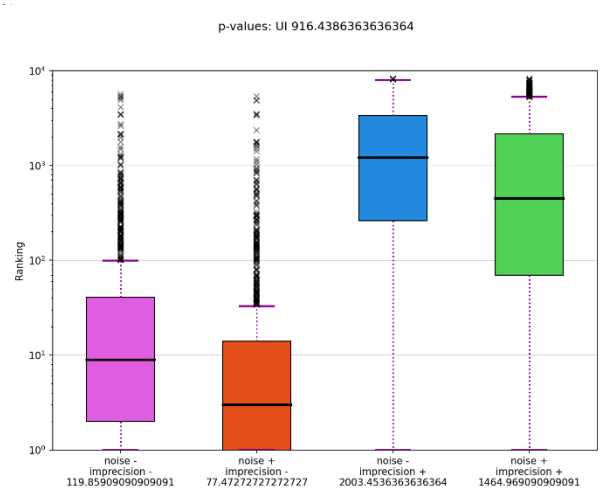
	Nº of 1st ranks	Medians
Noise - Imprecision -	0	26.0
Noise + Imprecision -	0	81.0
Noise - Imprecision +	0	20.0
Noise + Imprecision +	0	172.5

Normalized (Max) Term Overlap



	Nº of 1st ranks	Medians
Noise - Imprecision -	66	41.0
Noise + Imprecision -	199	13.0
Noise - Imprecision +	0	2922.5
Noise + Imprecision +	3	1150.5

Sim UI

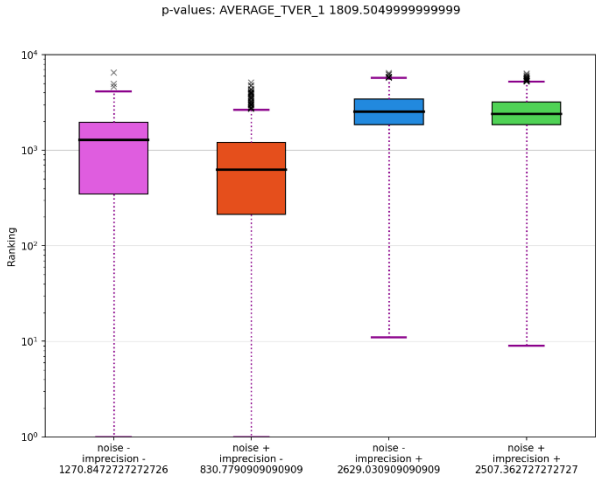
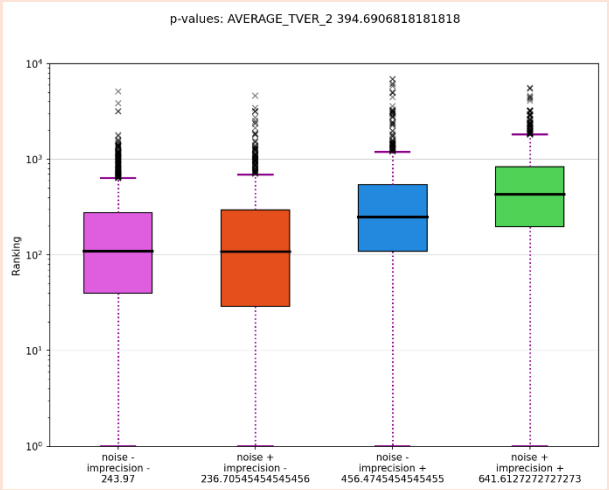


	Nº of 1st ranks	Medians
Noise - Imprecision -	227	9.0
Noise + Imprecision -	412	3.0
Noise - Imprecision +	5	1209.0
Noise + Imprecision +	14	453.5

Table 34. Direct Groupwise measures by p-values using the latest databases

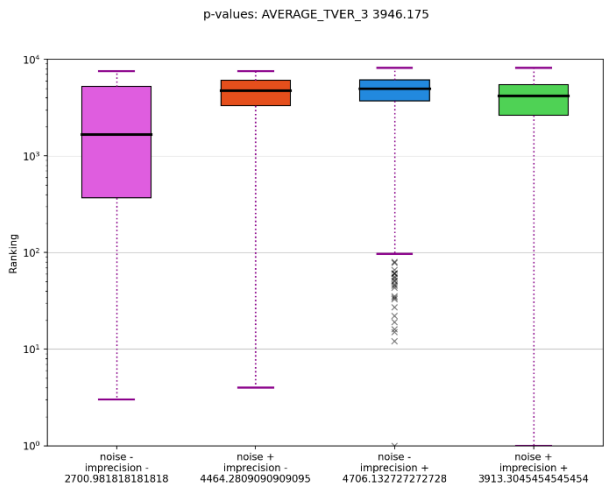
IC-based measures

IC-based measures by p-values

Method	Boxplot	Boxplot performance		
Pairwise: Tversky IC Contrast model **Indirect measure:** Average			**Nº of 1st ranks**	**Medians**
		Noise - Imprecision -	1	1282.5
		Noise + Imprecision -	3	625.0
		Noise - Imprecision +	0	2544.5
		Noise + Imprecision +	0	2406.5
Pairwise: Tversky IC Ratio model **Indirect measure:** Average			**Nº of 1st ranks**	**Medians**
		Noise - Imprecision -	9	110.0
		Noise + Imprecision -	25	108.0
		Noise - Imprecision +	1	249.5
		Noise + Imprecision +	2	429.0

Pairwise: Tversky IC Prop Contrast model

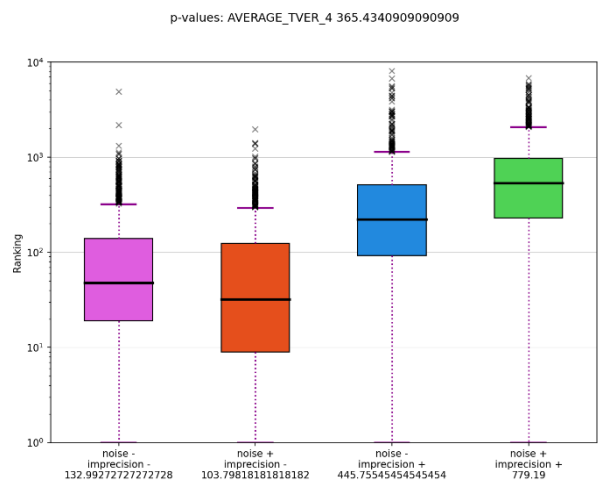
Indirect measure: Average



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	1671.5
Noise + Imprecision -	0	4779.5
Noise - Imprecision +	1	4979.5
Noise + Imprecision +	1	4193.5

Pairwise: Tversky IC Prop Ratio model

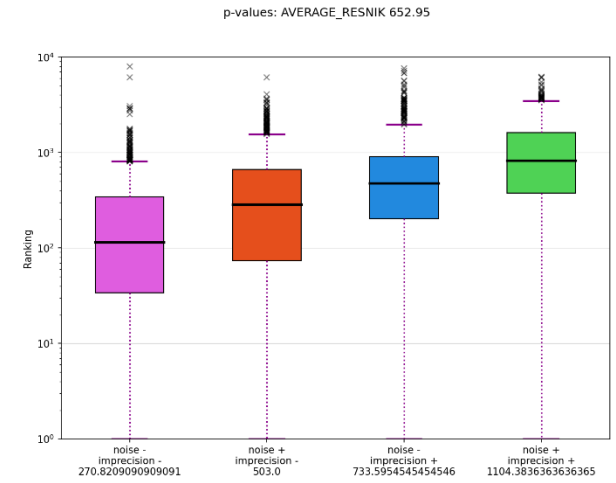
Indirect measure: Average



	Nº of 1st ranks	Medians
Noise - Imprecision -	13	48.0
Noise + Imprecision -	66	32.0
Noise - Imprecision +	1	223.5
Noise + Imprecision +	3	534.5

Pairwise: Resnik 1995

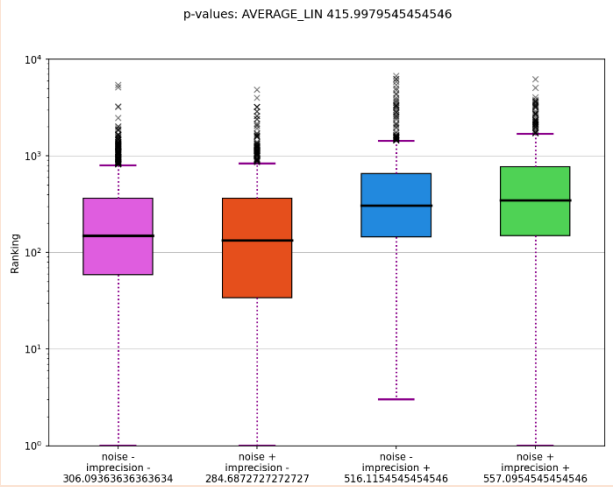
Indirect measure: Average



	Nº of 1st ranks	Medians
Noise - Imprecision -	12	115.5
Noise + Imprecision -	20	286.0
Noise - Imprecision +	3	479.0
Noise + Imprecision +	4	820.5

Pairwise: Lin 1998

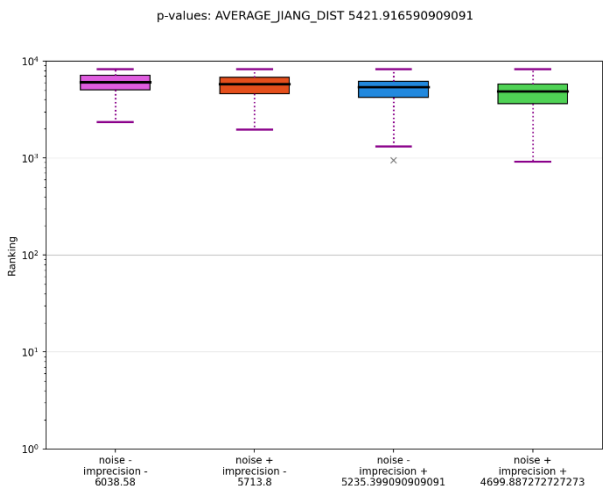
Indirect measure: Average



	Nº of 1st ranks	Medians
Noise - Imprecision -	1	149.5
Noise + Imprecision -	30	133.0
Noise - Imprecision +	0	306.5
Noise + Imprecision +	2	346.5

Pairwise: Jiang and Conrath 1997

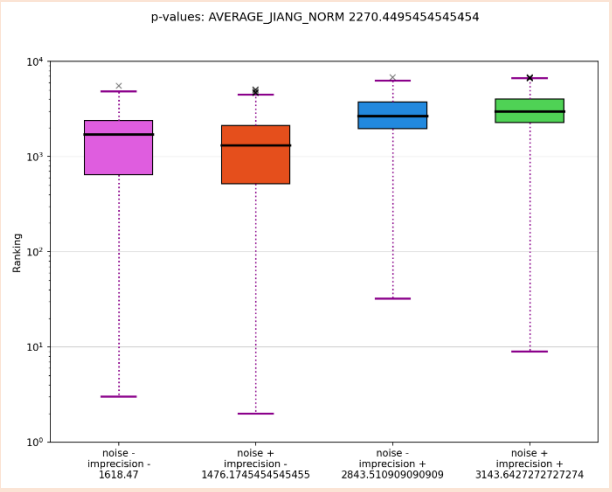
Indirect measure: Average



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	6029.5
Noise + Imprecision -	0	5768.0
Noise - Imprecision +	0	5427.5
Noise + Imprecision +	0	4884.5

Pairwise: Jiang and Conrath 1997 Normalized

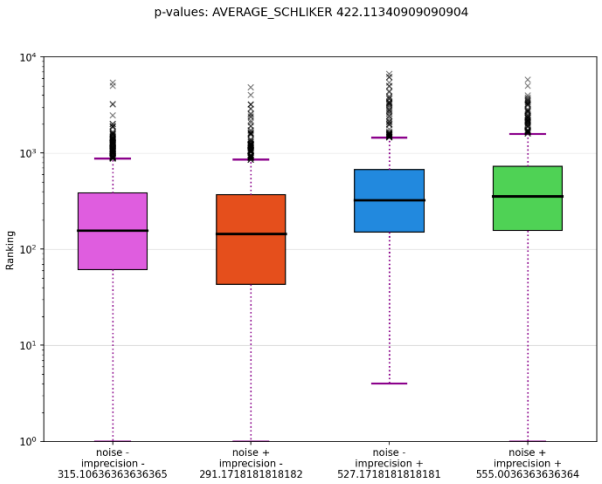
Indirect measure: Average



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	1702.0
Noise + Imprecision -	0	1320.5
Noise - Imprecision +	0	2670.5
Noise + Imprecision +	0	2986.5

Pairwise: Schlicker SimRel 2006

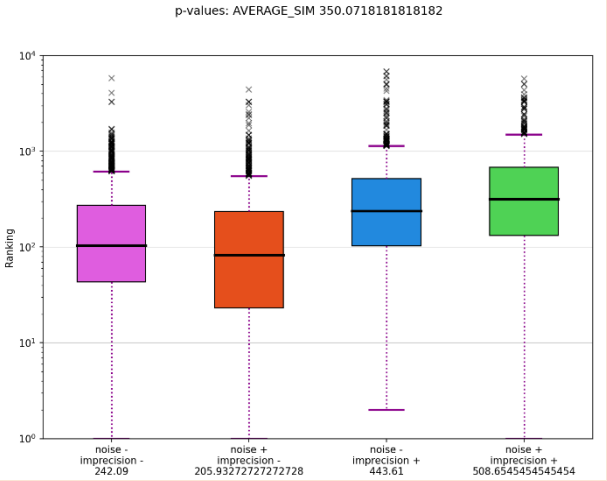
Indirect measure: Average



	Nº of 1st ranks	Medians
Noise - Imprecision -	12	156.0
Noise + Imprecision -	14	144.5
Noise - Imprecision +	0	323.0
Noise + Imprecision +	1	356.5

Pairwise: SimIC 2010

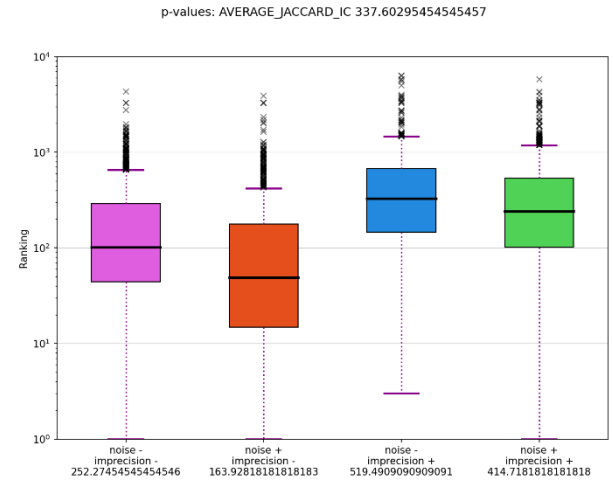
Indirect measure: Average



	Nº of 1st ranks	Medians
Noise - Imprecision -	9	104.0
Noise + Imprecision -	19	83.0
Noise - Imprecision +	0	239.5
Noise + Imprecision +	1	317.5

Pairwise: Jaccard IC

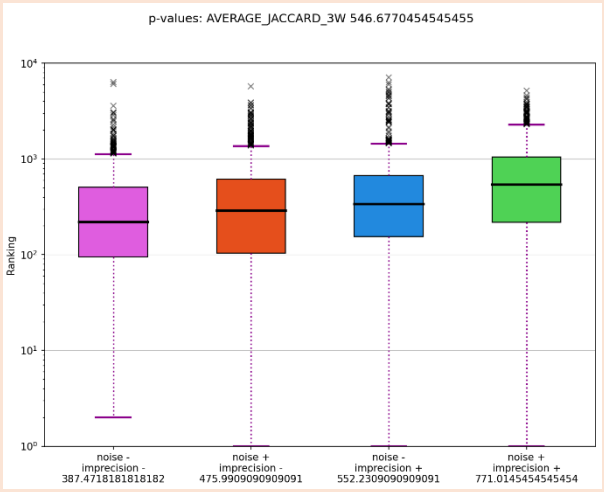
Indirect measure: Average



	Nº of 1st ranks	Medians
Noise - Imprecision -	10	101.0
Noise + Imprecision -	44	49.0
Noise - Imprecision +	0	329.0
Noise + Imprecision +	2	241.0

Pairwise: Jaccard 3W IC

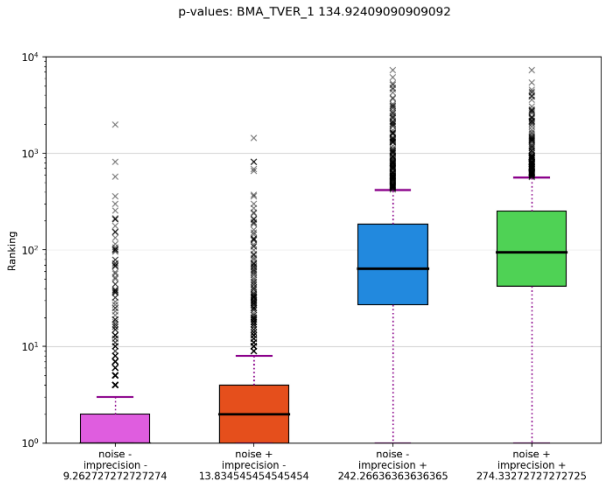
Indirect measure: Average



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	221.0
Noise + Imprecision -	16	289.0
Noise - Imprecision +	2	338.5
Noise + Imprecision +	4	543.5

Pairwise: Tversky IC Contrast model

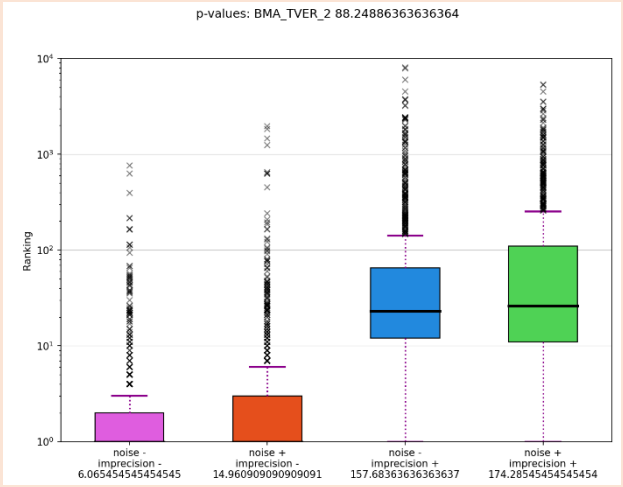
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	724	1.0
Noise + Imprecision -	545	2.0
Noise - Imprecision +	11	64.0
Noise + Imprecision +	6	95.0

Pairwise: Tversky IC Ratio model

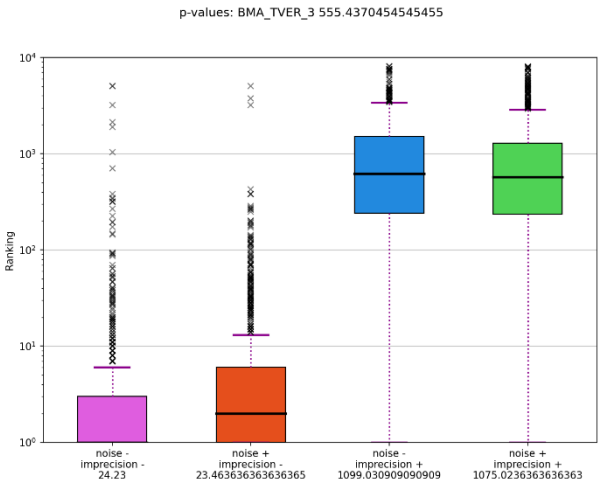
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	763	1.0
Noise + Imprecision -	624	1.0
Noise - Imprecision +	27	23.0
Noise + Imprecision +	12	26.0

Pairwise: Tversky IC Prop Contrast model

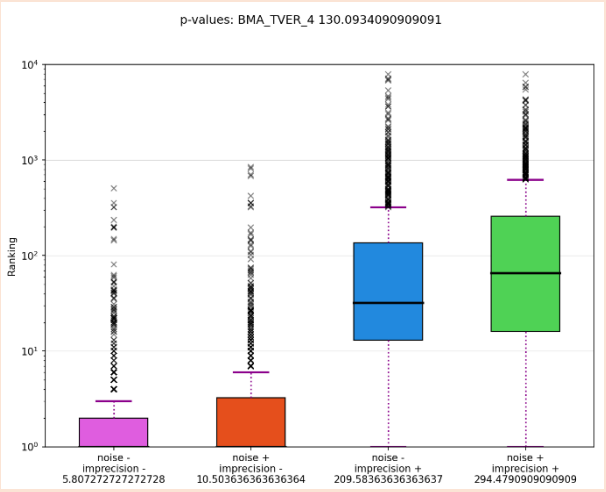
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	677	1.0
Noise + Imprecision -	411	2.0
Noise - Imprecision +	5	618.0
Noise + Imprecision +	3	576.0

Pairwise: Tversky IC Prop Ratio model

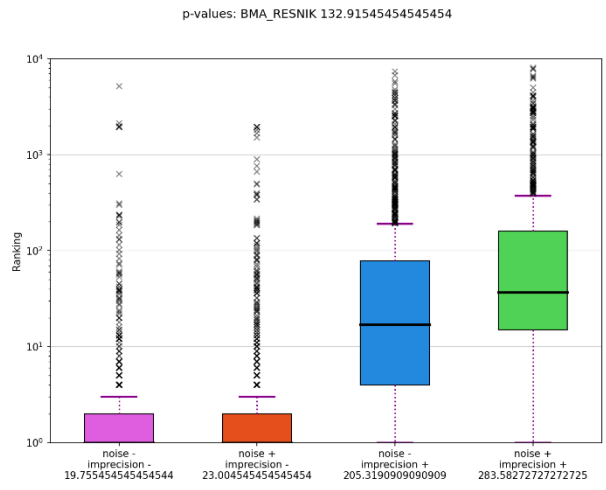
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	796	1.0
Noise + Imprecision -	565	1.0
Noise - Imprecision +	21	32.0
Noise + Imprecision +	15	66.0

Pairwise: Resnik 1995

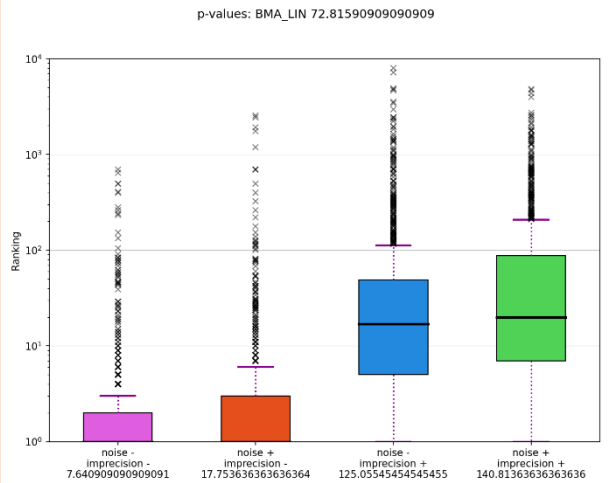
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	769	1.0
Noise + Imprecision -	686	1.0
Noise - Imprecision +	122	17.0
Noise + Imprecision +	55	37.0

Pairwise: Lin 1998

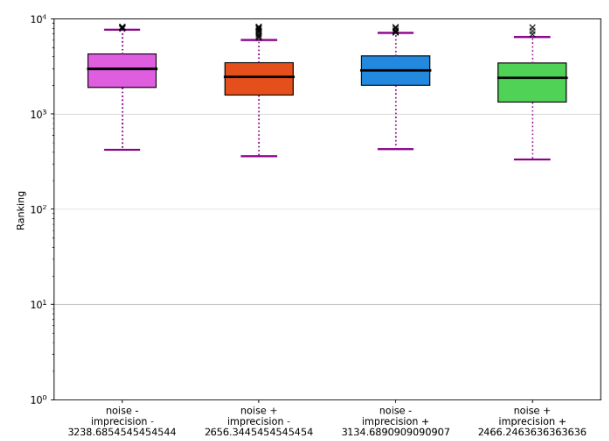
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	753	1.0
Noise + Imprecision -	660	1.0
Noise - Imprecision +	101	17.0
Noise + Imprecision +	83	20.0

Pairwise: Jiang and Conrath 1997

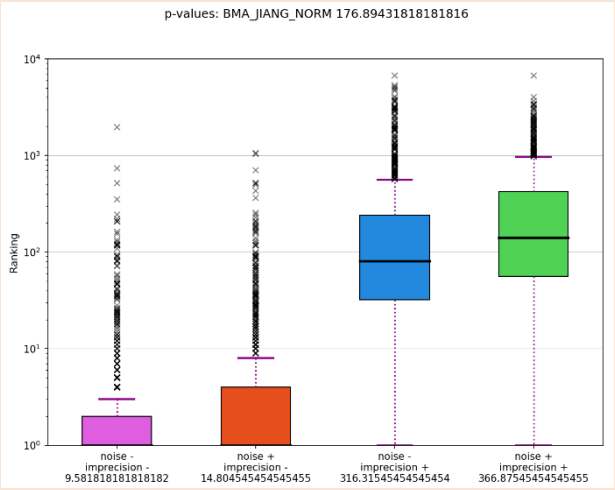
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	2977.5
Noise + Imprecision -	0	2461.5
Noise - Imprecision +	0	2873.5
Noise + Imprecision +	0	2405.0

Pairwise: Jiang and Conrath 1997 Normalized

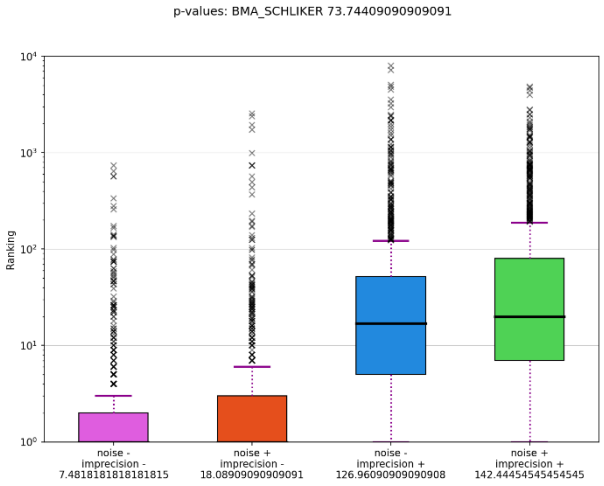
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	745	1.0
Noise + Imprecision -	604	1.0
Noise - Imprecision +	9	81.0
Noise + Imprecision +	3	140.5

Pairwise: Schlicker SimRel 2006

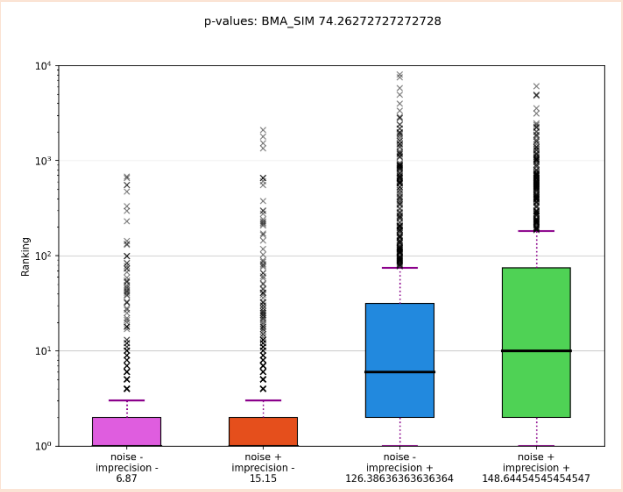
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	754	1.0
Noise + Imprecision -	655	1.0
Noise - Imprecision +	101	17.0
Noise + Imprecision +	81	20.0

Pairwise: SimIC 2010

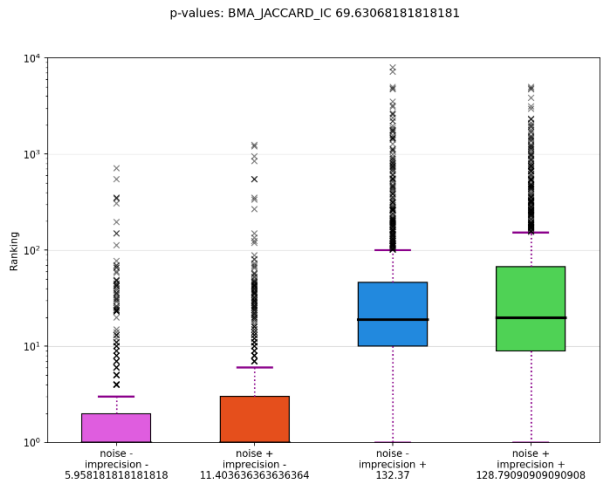
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	803	1.0
Noise + Imprecision -	692	1.0
Noise - Imprecision +	223	6.0
Noise + Imprecision +	190	10.0

Pairwise: Jaccard IC

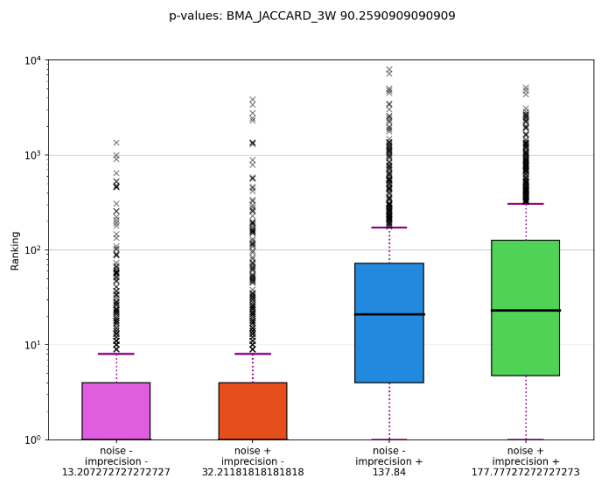
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	806	1.0
Noise + Imprecision -	605	1.0
Noise - Imprecision +	41	19.0
Noise + Imprecision +	22	20.0

Pairwise: Jaccard 3W IC

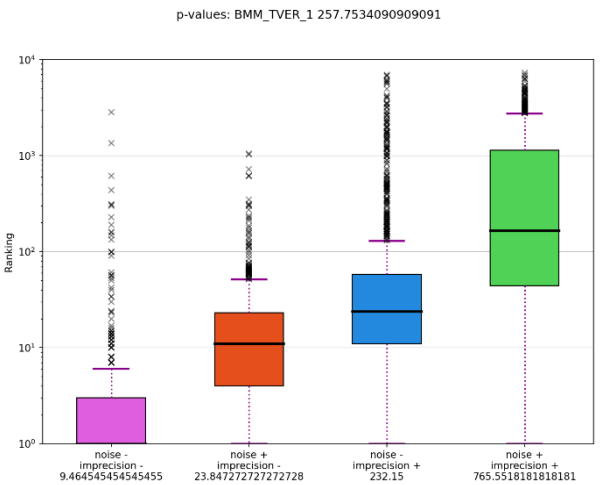
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians 1st ranks
Noise - Imprecision -	645	1.0
Noise + Imprecision -	649	1.0
Noise - Imprecision +	129	21.0
Noise + Imprecision +	138	23.0

Pairwise: Tversky IC Contrast model

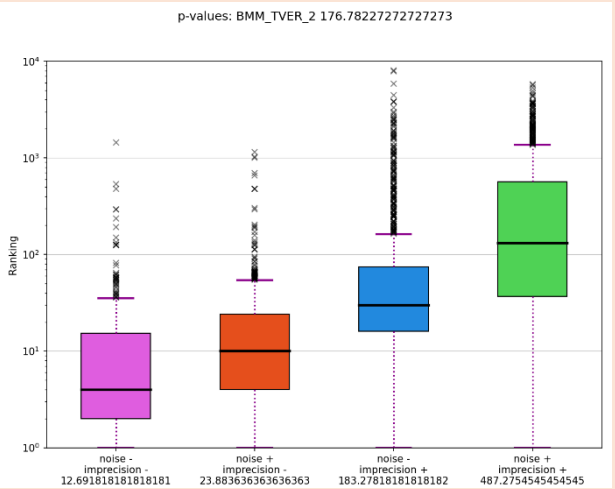
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians 1st ranks
Noise - Imprecision -	621	1.0
Noise + Imprecision -	71	11.0
Noise - Imprecision +	40	24.0
Noise + Imprecision +	3	166.0

Pairwise: Tversky IC Ratio model

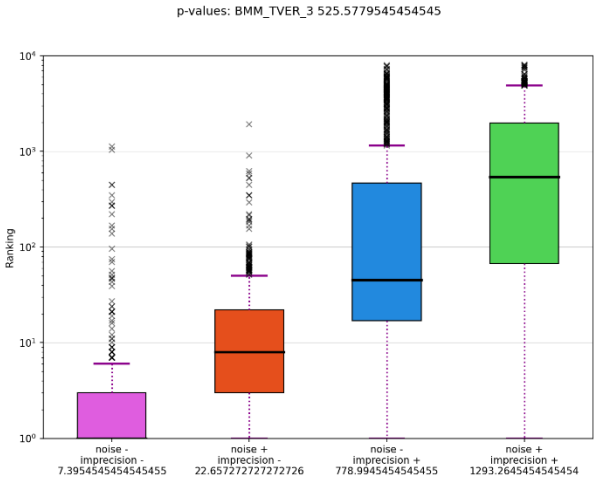
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians 1st ranks
Noise - Imprecision -	239	4.0
Noise + Imprecision -	71	10.0
Noise - Imprecision +	10	30.0
Noise + Imprecision +	2	132.0

Pairwise: Tversky IC Prop Contrast model

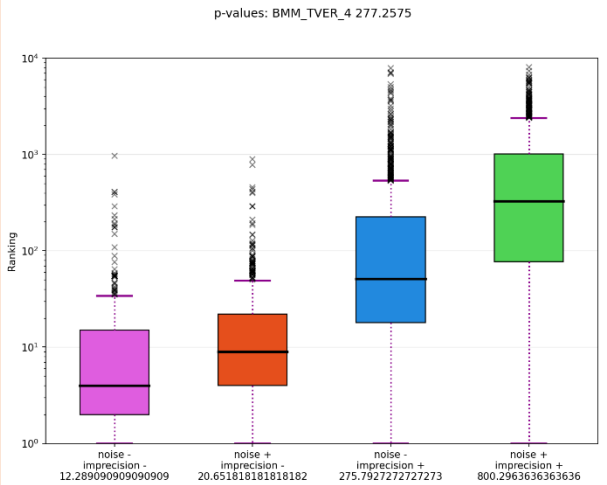
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians
Noise - Imprecision -	622	1.0
Noise + Imprecision -	93	8.0
Noise - Imprecision +	22	45.0
Noise + Imprecision +	3	540.0

Pairwise: Tversky IC Prop Ratio model

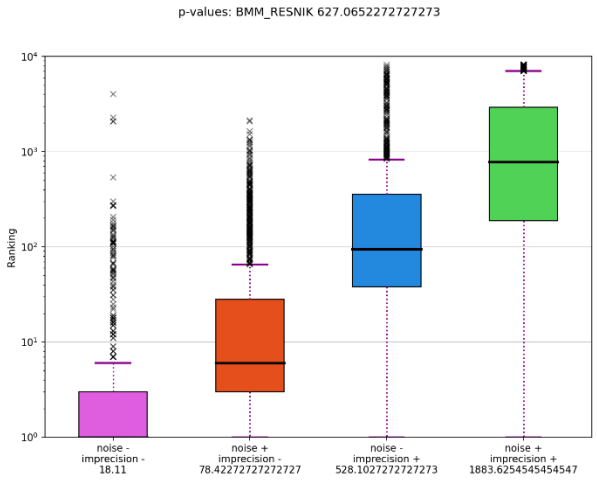
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians
Noise - Imprecision -	247	4.0
Noise + Imprecision -	82	9.0
Noise - Imprecision +	15	51.0
Noise + Imprecision +	5	329.0

Pairwise: Resnik 1995

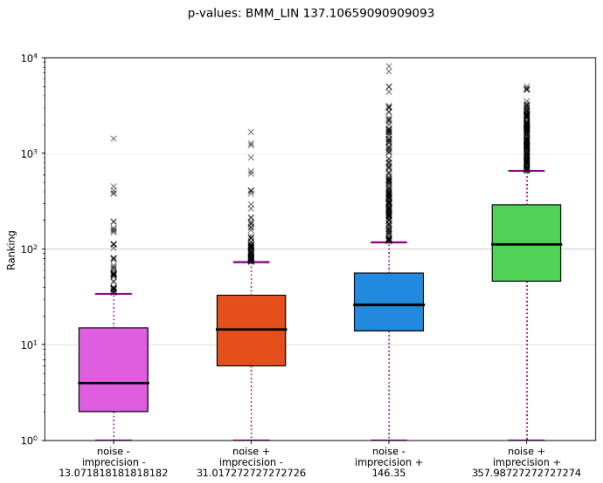
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians
Noise - Imprecision -	590	1.0
Noise + Imprecision -	106	6.0
Noise - Imprecision +	50	95.0
Noise + Imprecision +	5	781.0

Pairwise: Lin 1998

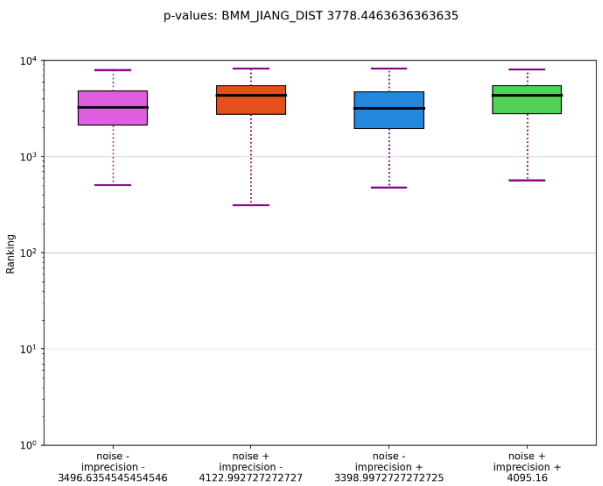
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians
Noise - Imprecision -	245	4.0
Noise + Imprecision -	54	14.5
Noise - Imprecision +	11	26.0
Noise + Imprecision +	2	113.0

Pairwise: Jiang and Conrath 1997

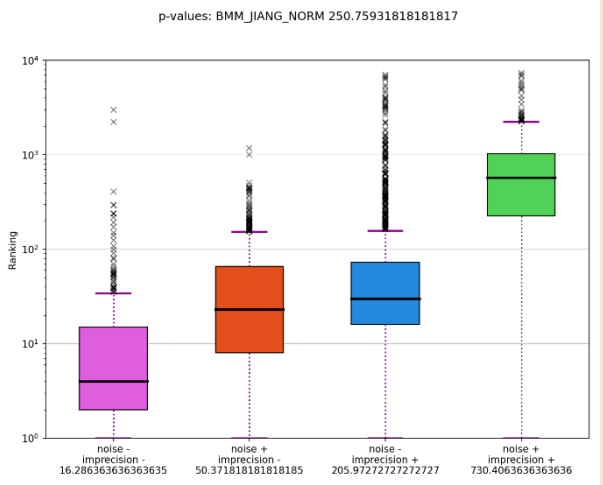
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	3285.0
Noise + Imprecision -	0	4355.0
Noise - Imprecision +	0	3199.5
Noise + Imprecision +	0	4357.0

Pairwise: Jiang and Conrath 1997 Normalized

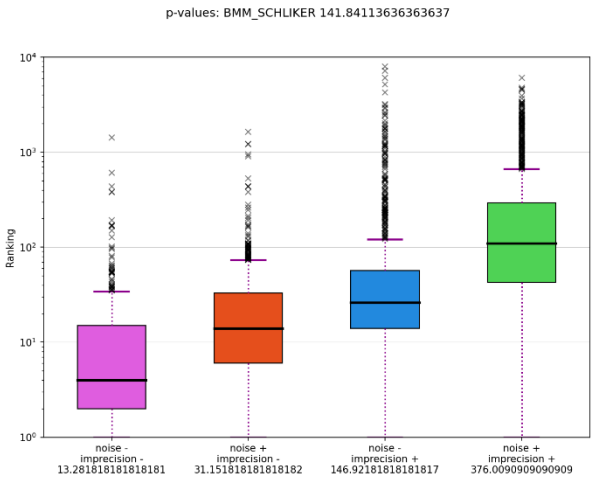
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians
Noise - Imprecision -	243	4.0
Noise + Imprecision -	50	23.0
Noise - Imprecision +	7	30.0
Noise + Imprecision +	2	572.5

Pairwise: Schlicker SimRel 2006

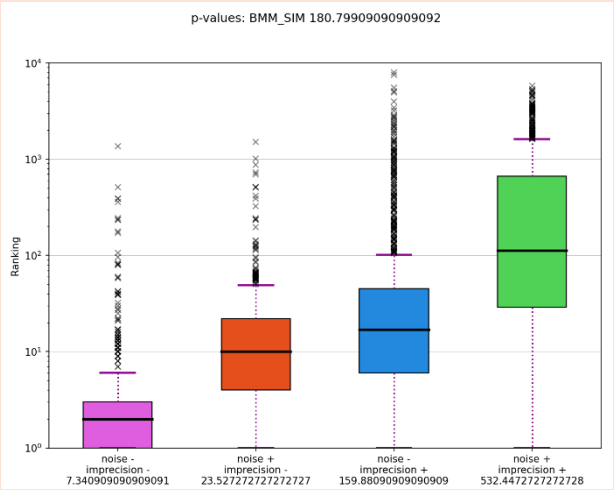
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians
Noise - Imprecision -	244	4.0
Noise + Imprecision -	57	14.0
Noise - Imprecision +	13	26.0
Noise + Imprecision +	2	110.0

Pairwise: SimIC 2010

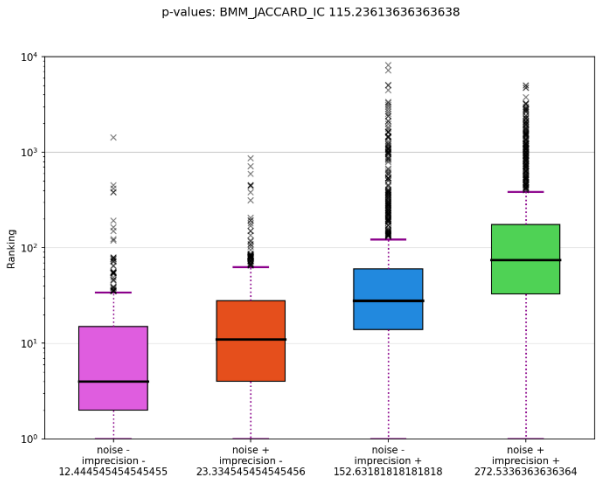
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians
Noise - Imprecision -	538	2.0
Noise + Imprecision -	74	10.0
Noise - Imprecision +	48	17.0
Noise + Imprecision +	5	112.5

Pairwise: Jaccard IC

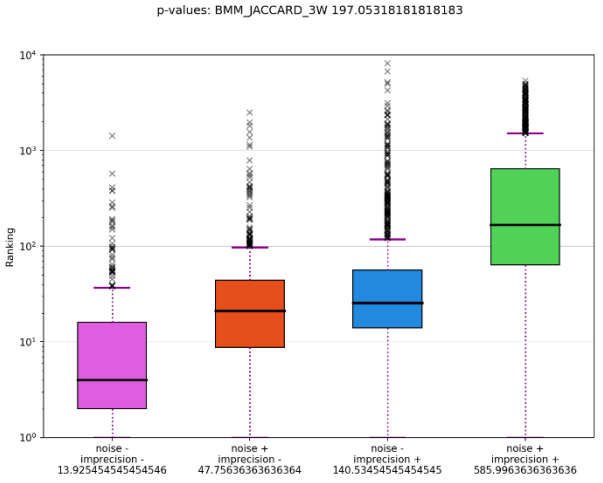
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians
Noise - Imprecision -	247	4.0
Noise + Imprecision -	73	11.0
Noise - Imprecision +	15	28.0
Noise + Imprecision +	2	74.5

Pairwise: Jaccard 3W IC

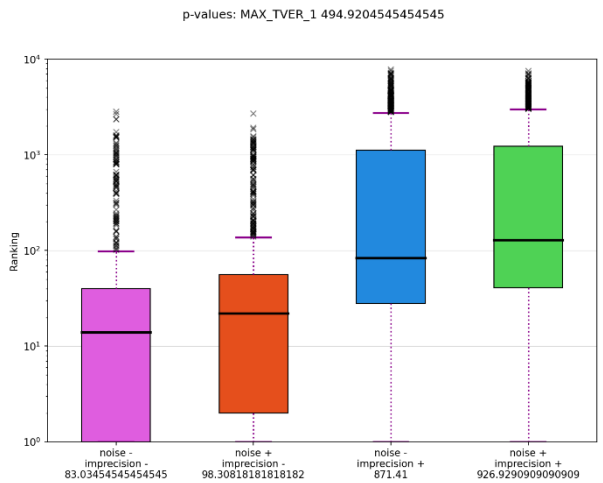
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians
Noise - Imprecision -	243	4.0
Noise + Imprecision -	41	21.0
Noise - Imprecision +	15	25.5
Noise + Imprecision +	1	167.0

Pairwise: Tversky IC Contrast model

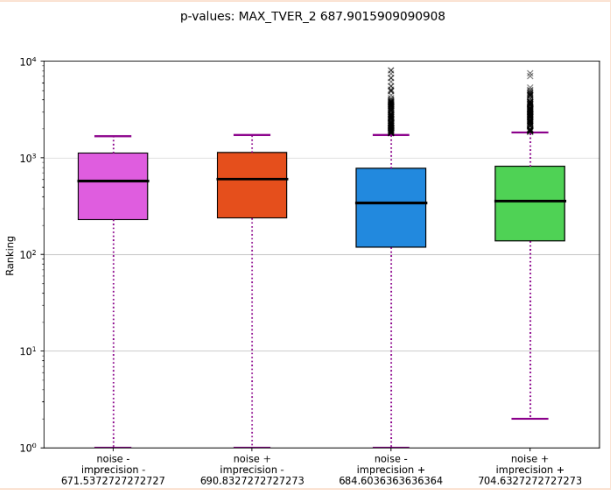
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	377	14.0
Noise + Imprecision -	246	22.0
Noise - Imprecision +	63	84.0
Noise + Imprecision +	41	128.5

Pairwise: Tversky IC Ratio model

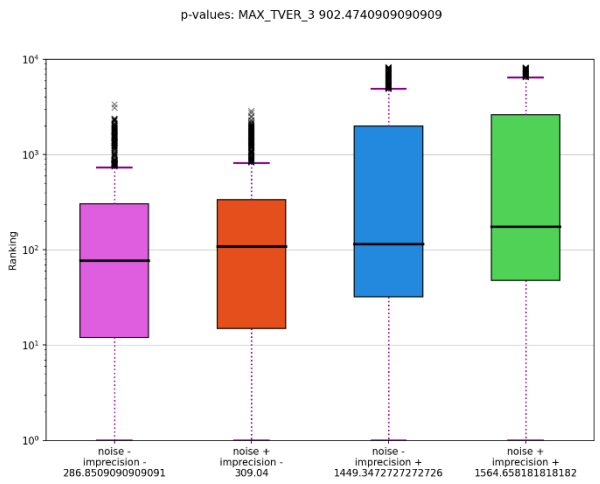
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	10	580.5
Noise + Imprecision -	8	609.5
Noise - Imprecision +	2	342.5
Noise + Imprecision +	0	358.0

Pairwise: Tversky IC Prop Contrast model

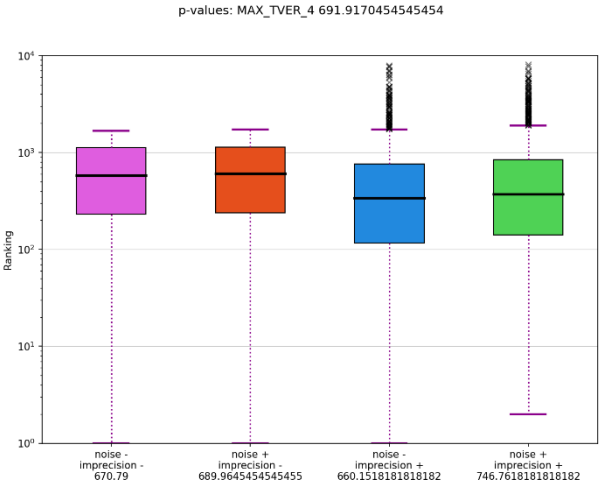
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	116	77.0
Noise + Imprecision -	101	108.5
Noise - Imprecision +	9	114.5
Noise + Imprecision +	12	175.0

Pairwise: Tversky IC Prop Ratio model

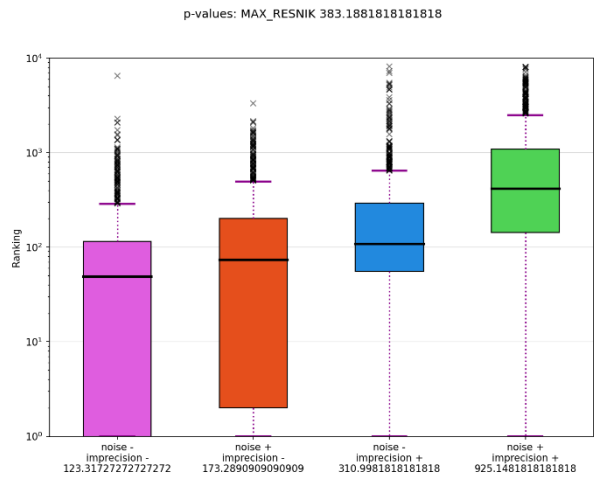
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	10	580.5
Noise + Imprecision -	8	608.5
Noise - Imprecision +	3	338.0
Noise + Imprecision +	0	373.0

Pairwise: Resnik 1995

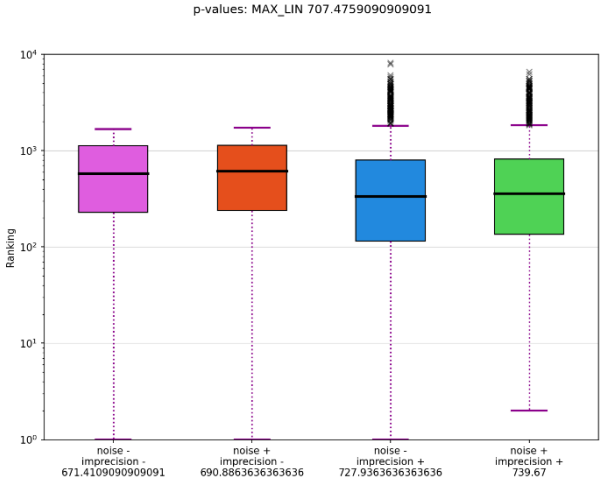
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	389	49.0
Noise + Imprecision -	246	74.0
Noise - Imprecision +	58	108.5
Noise + Imprecision +	42	417.0

Pairwise: Lin 1998

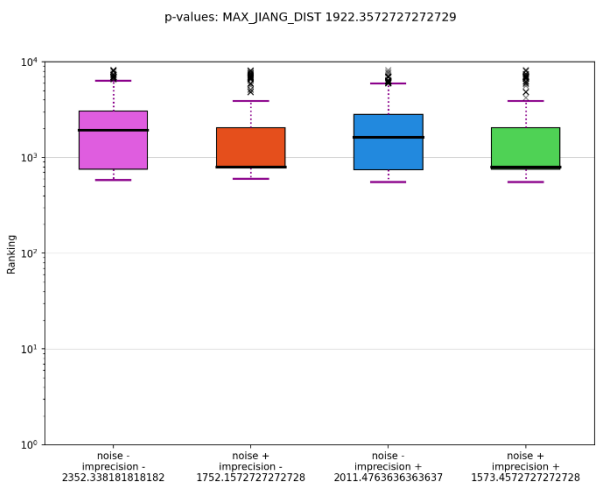
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	10	580.5
Noise + Imprecision -	8	613.5
Noise - Imprecision +	2	334.5
Noise + Imprecision +	0	358.5

Pairwise: Jiang and Conrath 1997

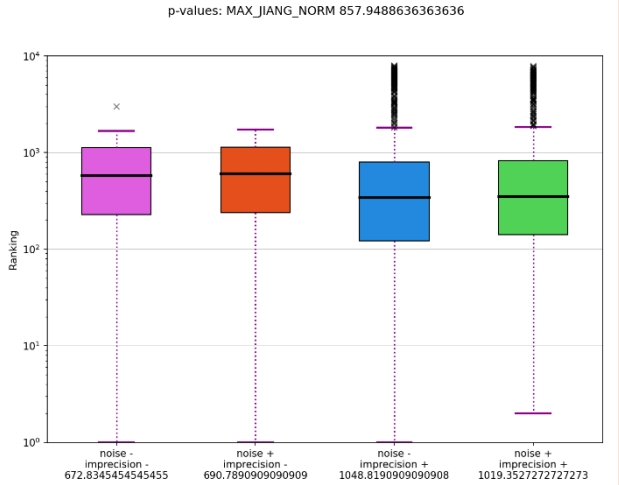
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	1937.0
Noise + Imprecision -	0	793.0
Noise - Imprecision +	0	1627.0
Noise + Imprecision +	0	793.0

Pairwise: Jiang and Conrath 1997 Normalized

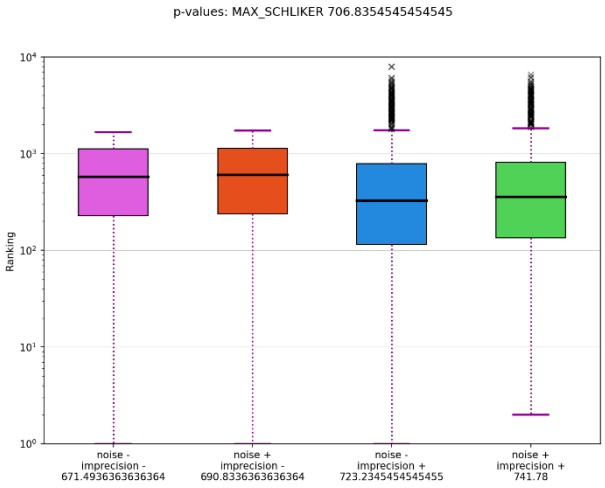
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	10	580.5
Noise + Imprecision -	8	609.5
Noise - Imprecision +	2	344.5
Noise + Imprecision +	0	350.0

Pairwise: Schlicker SimRel 2006

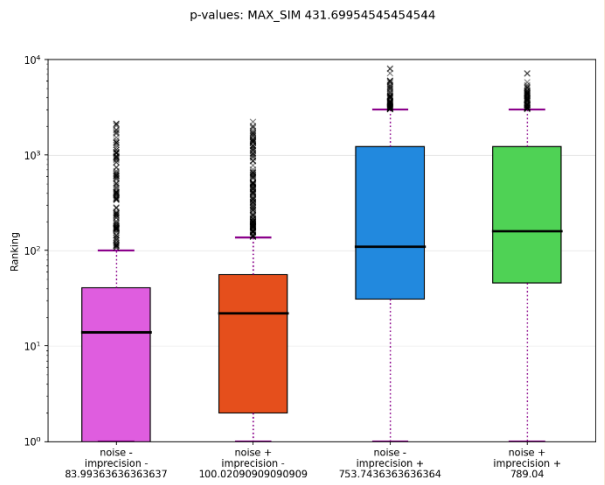
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	10	580.5
Noise + Imprecision -	8	609.5
Noise - Imprecision +	2	327.5
Noise + Imprecision +	0	358.5

Pairwise: SimIC 2010

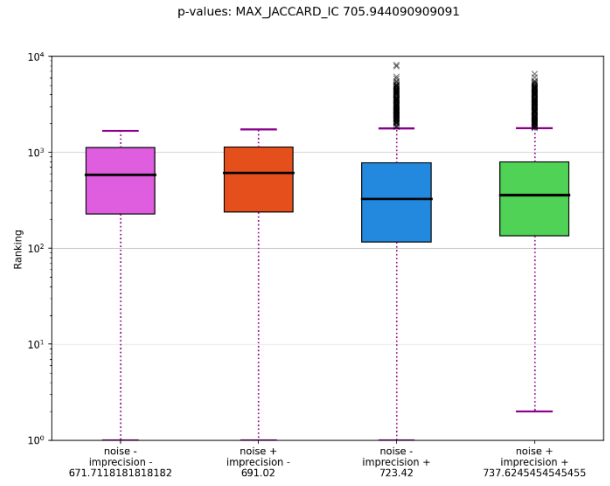
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	378	14.0
Noise + Imprecision -	252	22.0
Noise - Imprecision +	62	110.0
Noise + Imprecision +	42	160.0

Pairwise: Jaccard IC

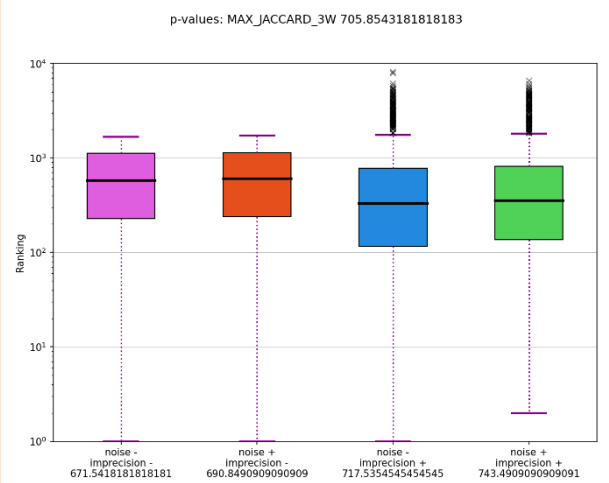
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	10	584.0
Noise + Imprecision -	8	612.0
Noise - Imprecision +	2	327.5
Noise + Imprecision +	0	358.0

Pairwise: Jaccard 3W IC

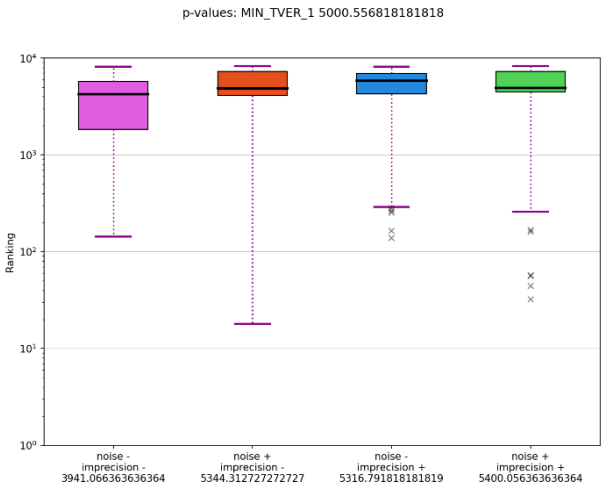
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	10	580.5
Noise + Imprecision -	8	609.5
Noise - Imprecision +	2	330.5
Noise + Imprecision +	0	357.0

Pairwise: Tversky IC Contrast model

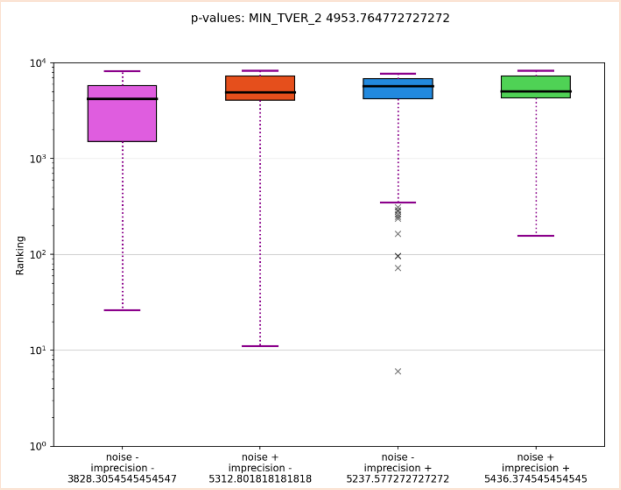
Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	4268.0
Noise + Imprecision -	0	4892.0
Noise - Imprecision +	0	5861.0
Noise + Imprecision +	0	4916.0

Pairwise: Tversky IC Ratio model

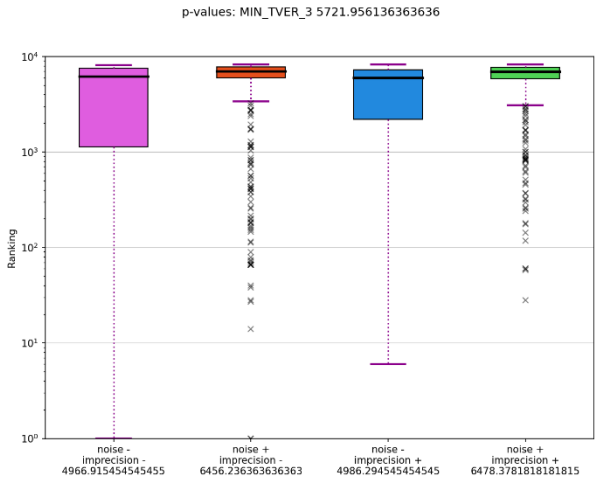
Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	4193.5
Noise + Imprecision -	0	4913.0
Noise - Imprecision +	0	5713.0
Noise + Imprecision +	0	5033.0

Pairwise: Tversky IC Prop Contrast model

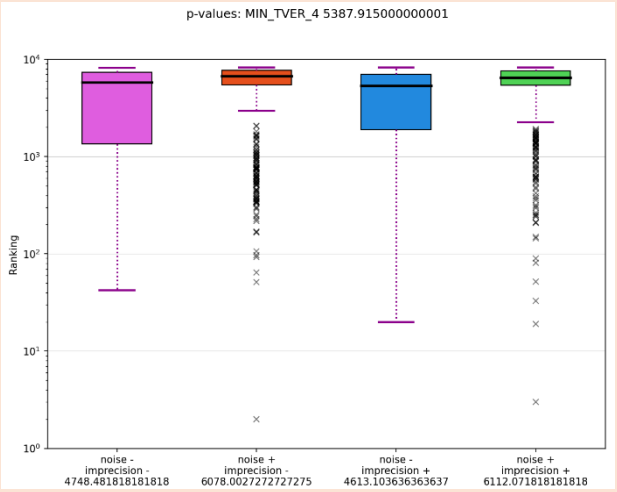
Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	2	6185.0
Noise + Imprecision -	2	7015.0
Noise - Imprecision +	0	6012.0
Noise + Imprecision +	0	6916.5

Pairwise: Tversky IC Prop Ratio model

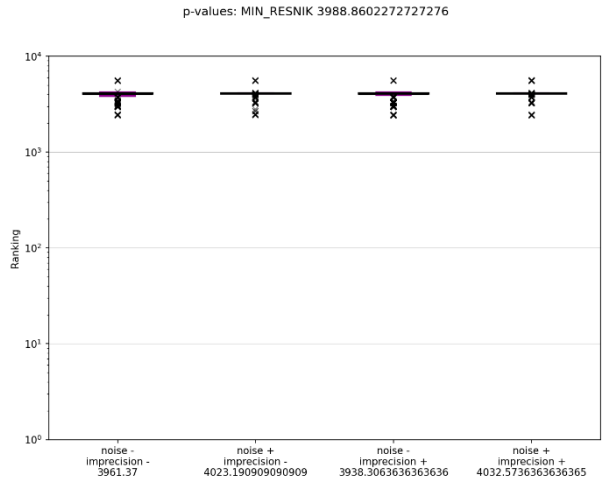
Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	5807.5
Noise + Imprecision -	0	6706.0
Noise - Imprecision +	0	5358.5
Noise + Imprecision +	0	6493.5

Pairwise: Resnik 1995

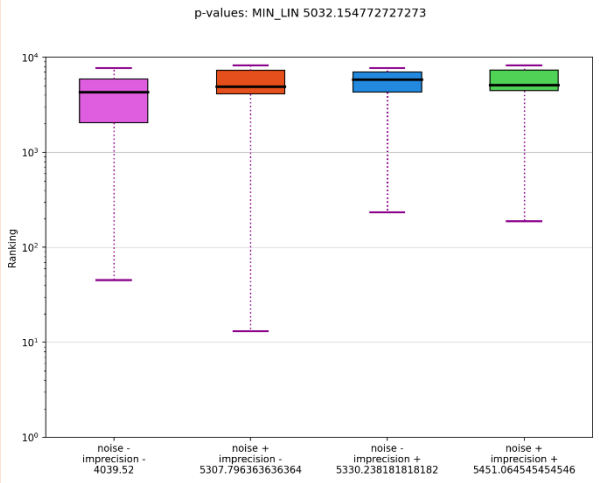
Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	4092.0
Noise + Imprecision -	0	4092.0
Noise - Imprecision +	0	4092.0
Noise + Imprecision +	0	4092.0

Pairwise: Lin 1998

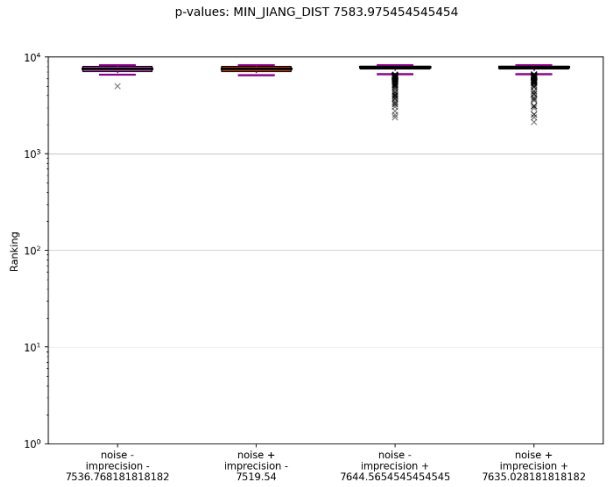
Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	4320.5
Noise + Imprecision -	0	4915.0
Noise - Imprecision +	0	5835.0
Noise + Imprecision +	0	5126.0

Pairwise: Jiang and Conrath 1997

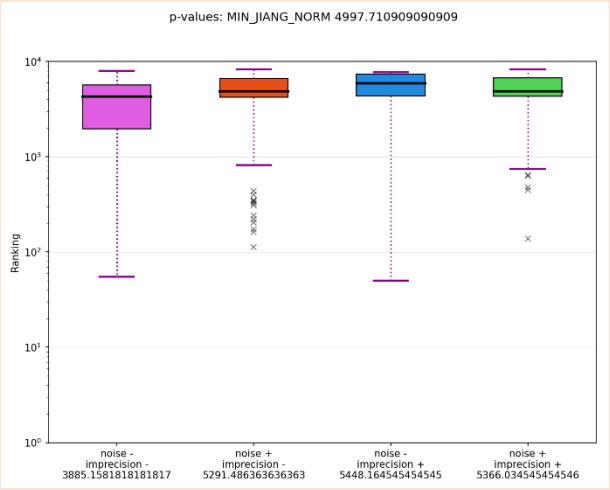
Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	7628.5
Noise + Imprecision -	0	7599.5
Noise - Imprecision +	0	7877.5
Noise + Imprecision +	0	7875.0

Pairwise: Jiang and Conrath 1997 Normalized

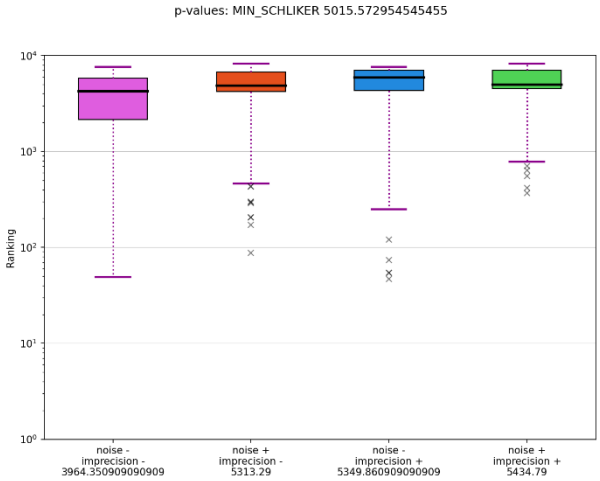
Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	4300.5
Noise + Imprecision -	0	4884.5
Noise - Imprecision +	0	5954.0
Noise + Imprecision +	0	4889.5

Pairwise: Schlicker SimRel 2006

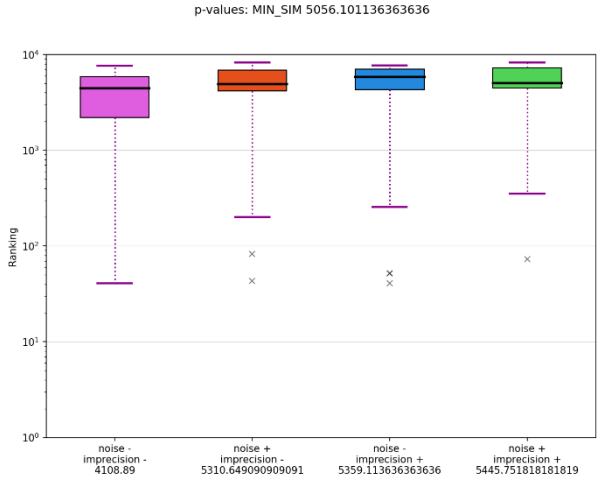
Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	4273.0
Noise + Imprecision -	0	4896.0
Noise - Imprecision +	0	5889.0
Noise + Imprecision +	0	4982.0

Pairwise: SimIC 2010

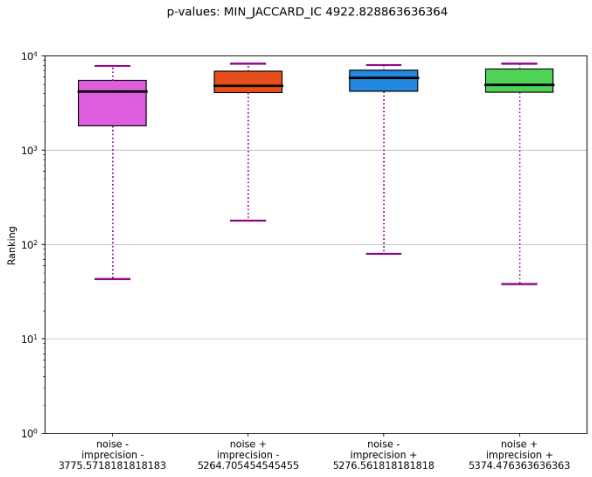
Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	4449.0
Noise + Imprecision -	0	4920.0
Noise - Imprecision +	0	5877.5
Noise + Imprecision +	0	5072.0

Pairwise: Jaccard IC

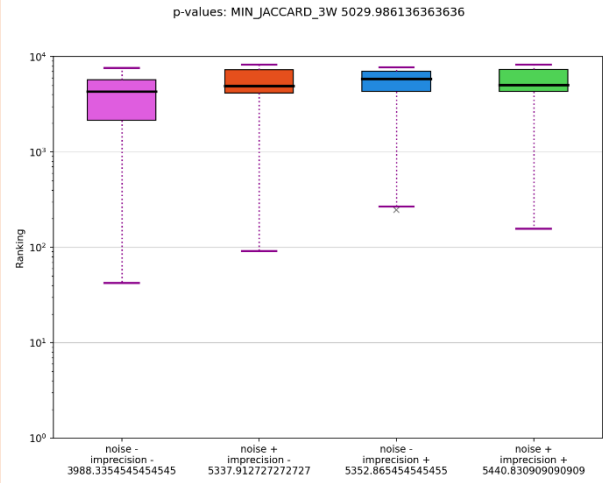
Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	4205.5
Noise + Imprecision -	0	4838.0
Noise - Imprecision +	0	5832.5
Noise + Imprecision +	0	4938.0

Pairwise: Jaccard 3W IC

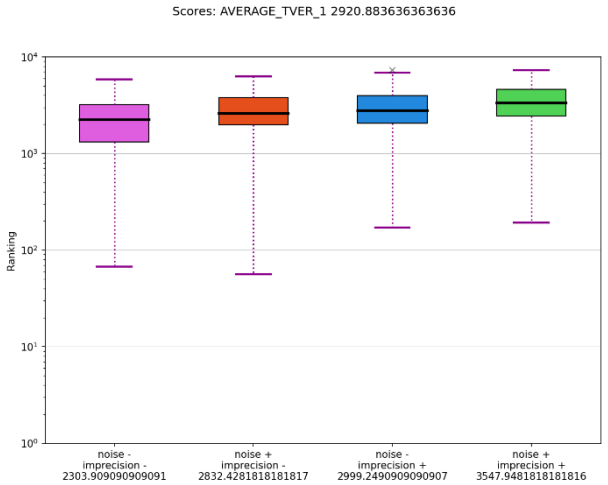
Indirect measure: Minimum



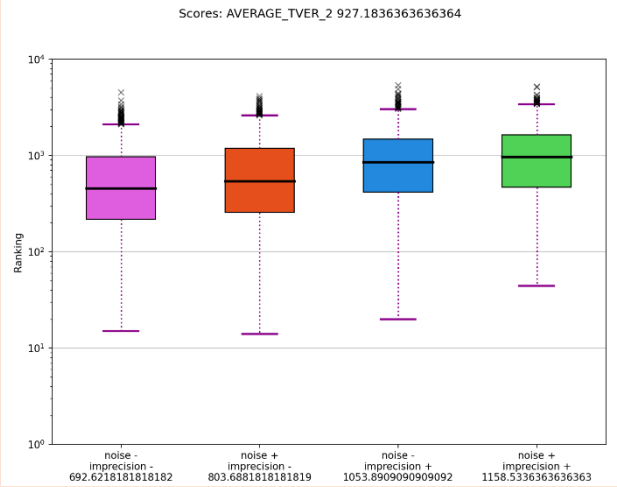
	Nº of 1st ranks	Medians
Noise - Imprecision -	0	4290.5
Noise + Imprecision -	0	4932.5
Noise - Imprecision +	0	5879.0
Noise + Imprecision +	0	5036.0

Table 35. IC-based measures by p-values using the latest databases

IC-based measures by similarity scores

Method	Boxplot	Boxplot performance
Pairwise: Tversky IC Contrast model **Indirect measure:** Average		

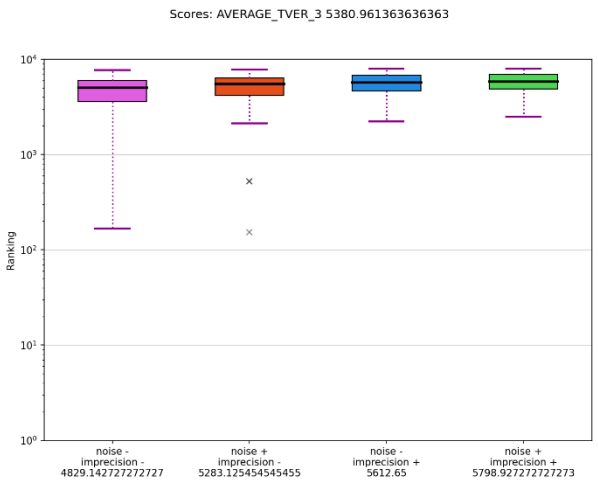
	Nº of 1st ranks	Medians
Noise - Imprecision -	0	2249.5
Noise + Imprecision -	0	2608.0
Noise - Imprecision +	0	2798.0
Noise + Imprecision +	0	3350.0

Method	Boxplot	Boxplot performance
Pairwise: Tversky IC Ratio model **Indirect measure:** Average		

	Nº of 1st ranks	Medians
Noise - Imprecision -	0	456.5
Noise + Imprecision -	0	542.0
Noise - Imprecision +	0	857.5
Noise + Imprecision +	0	967.0

Pairwise: Tversky IC Prop Contrast model

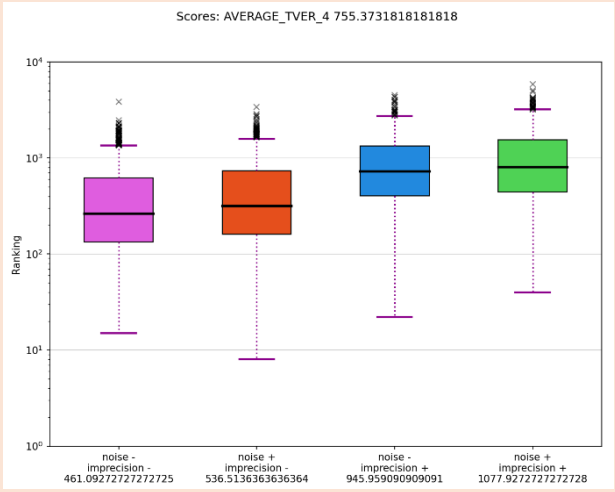
Indirect measure: Average



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	5043.0
Noise + Imprecision -	0	5522.5
Noise - Imprecision +	0	5700.5
Noise + Imprecision +	0	5875.0

Pairwise: Tversky IC Prop Ratio model

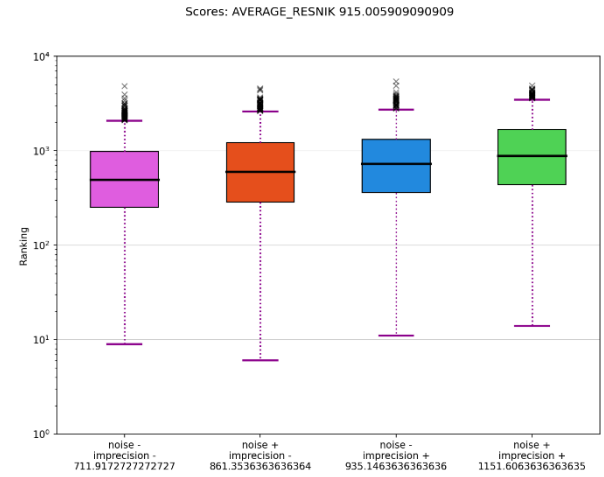
Indirect measure: Average



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	263.0
Noise + Imprecision -	0	318.0
Noise - Imprecision +	0	727.0
Noise + Imprecision +	0	802.0

Pairwise: Resnik 1995

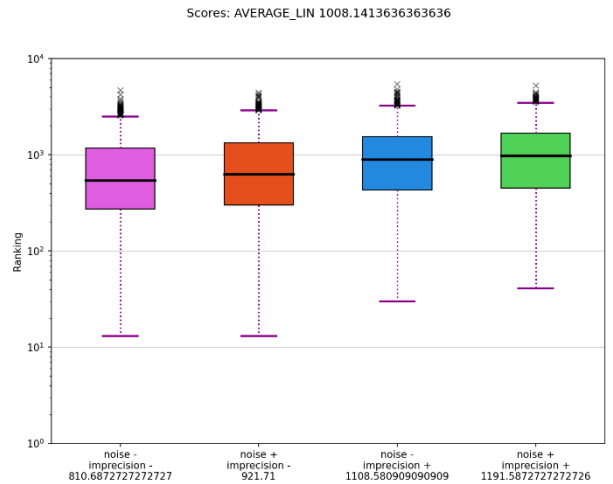
Indirect measure: Average



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	492.5
Noise + Imprecision -	0	598.0
Noise - Imprecision +	0	730.0
Noise + Imprecision +	0	887.5

Pairwise: Lin 1998

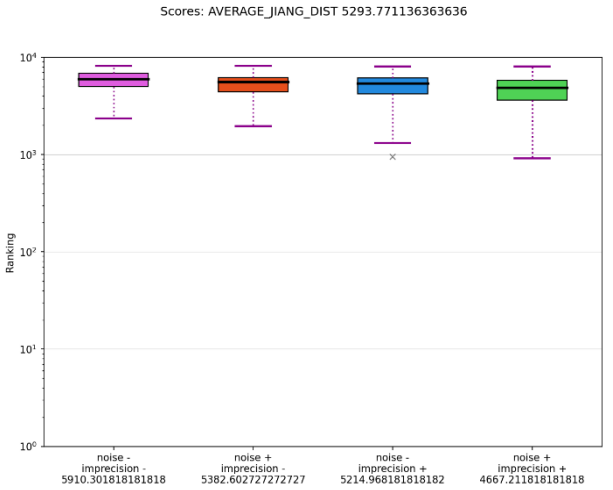
Indirect measure: Average



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	540.0
Noise + Imprecision -	0	630.0
Noise - Imprecision +	0	892.0
Noise + Imprecision +	0	976.0

Pairwise: Jiang and Conrath 1997

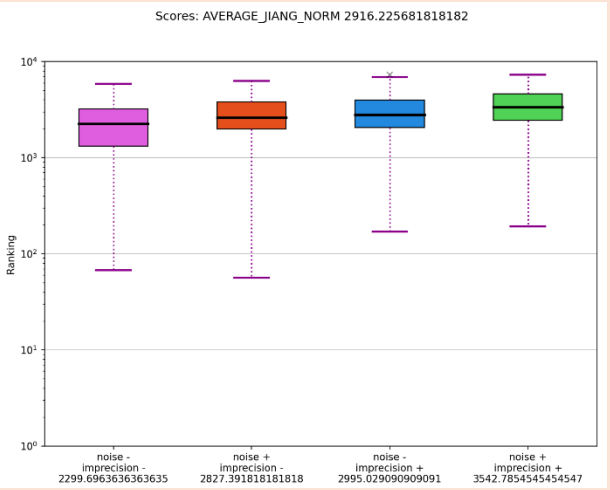
Indirect measure: Average



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	5964.5
Noise + Imprecision -	0	5607.5
Noise - Imprecision +	0	5416.5
Noise + Imprecision +	0	4867.0

Pairwise: Jiang and Conrath 1997 Normalized

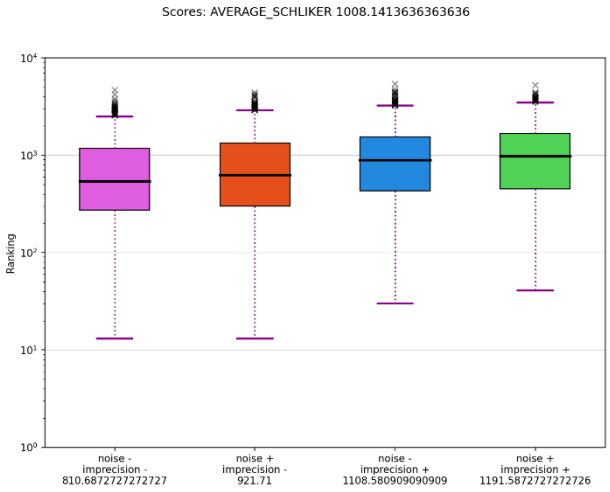
Indirect measure: Average



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	2245.5
Noise + Imprecision -	0	2602.5
Noise - Imprecision +	0	2793.5
Noise + Imprecision +	0	3343.0

Pairwise: Schlicker SimRel 2006

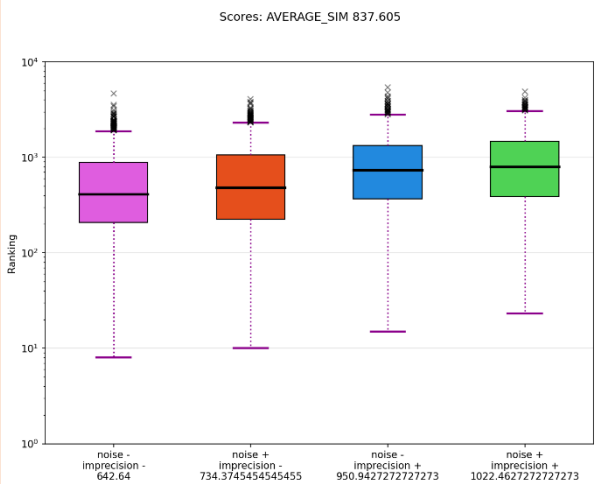
Indirect measure: Average



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	540.0
Noise + Imprecision -	0	630.0
Noise - Imprecision +	0	892.0
Noise + Imprecision +	0	976.0

Pairwise: SimIC 2010

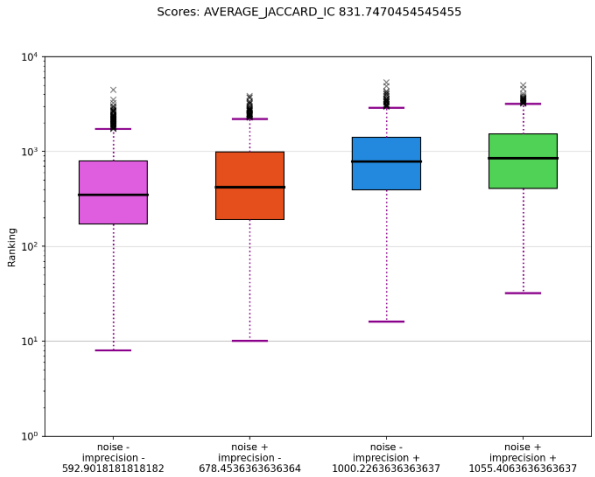
Indirect measure: Average



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	409.5
Noise + Imprecision -	0	481.5
Noise - Imprecision +	0	740.0
Noise + Imprecision +	0	799.5

Pairwise: Jaccard IC

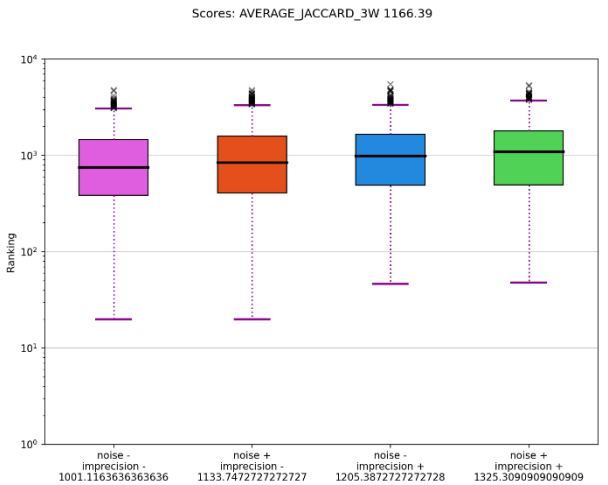
Indirect measure: Average



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	351.5
Noise + Imprecision -	0	419.5
Noise - Imprecision +	0	790.5
Noise + Imprecision +	0	850.0

Pairwise: Jaccard 3W IC

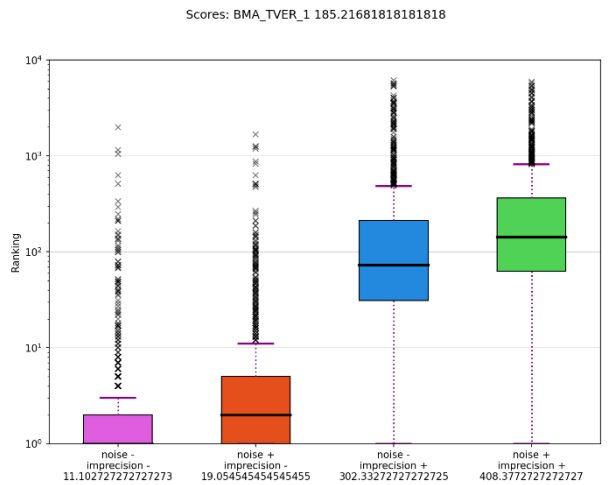
Indirect measure: Average



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	756.5
Noise + Imprecision -	0	844.5
Noise - Imprecision +	0	987.5
Noise + Imprecision +	0	1101.5

Pairwise: Tversky IC Contrast model

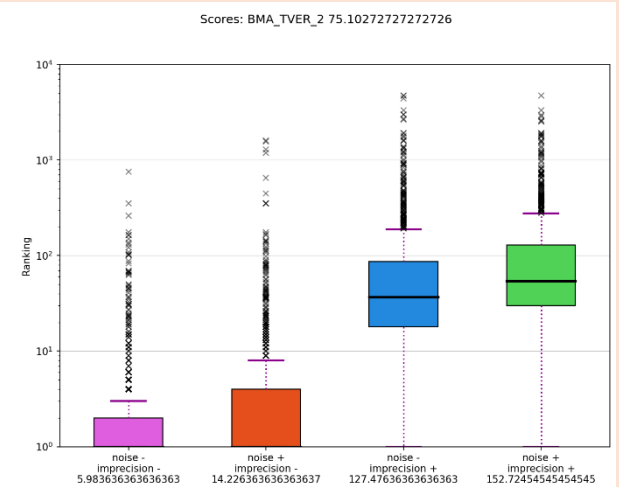
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	719	1.0
Noise + Imprecision -	475	2.0
Noise - Imprecision +	11	73.0
Noise + Imprecision +	3	142.5

Pairwise: Tversky IC Ratio model

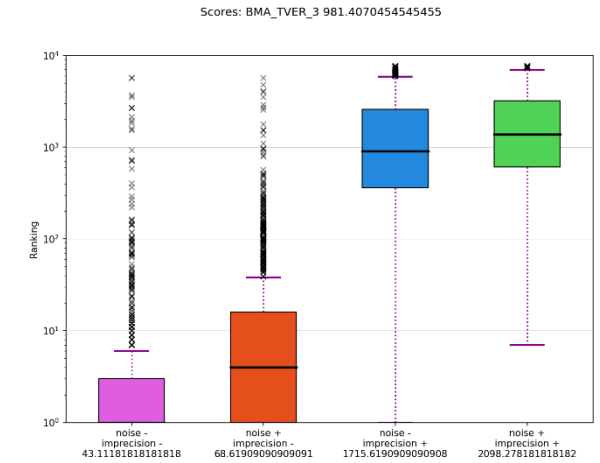
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	757	1.0
Noise + Imprecision -	585	1.0
Noise - Imprecision +	21	37.0
Noise + Imprecision +	3	54.0

Pairwise: Tversky IC Prop Contrast model

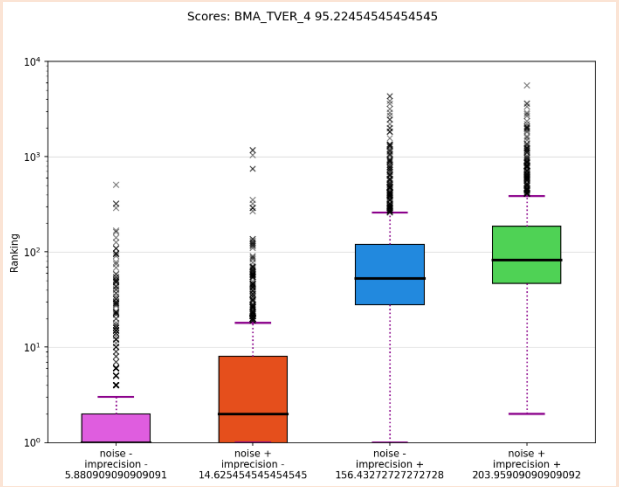
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	641	1.0
Noise + Imprecision -	314	4.0
Noise - Imprecision +	4	903.0
Noise + Imprecision +	0	1380.0

Pairwise: Tversky IC Prop Ratio model

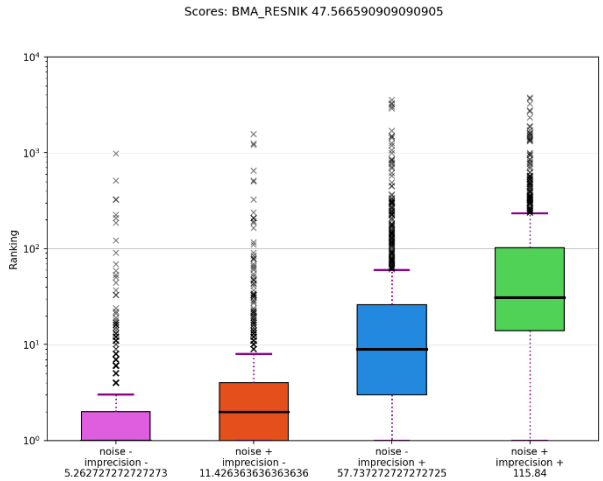
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	787	1.0
Noise + Imprecision -	435	2.0
Noise - Imprecision +	5	53.0
Noise + Imprecision +	0	83.0

Pairwise: Resnik 1995

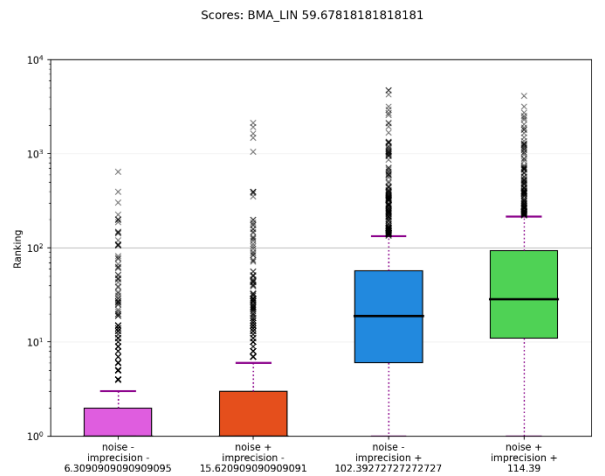
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	704	1.0
Noise + Imprecision -	381	2.0
Noise - Imprecision +	114	9.0
Noise + Imprecision +	20	31.0

Pairwise: Lin 1998

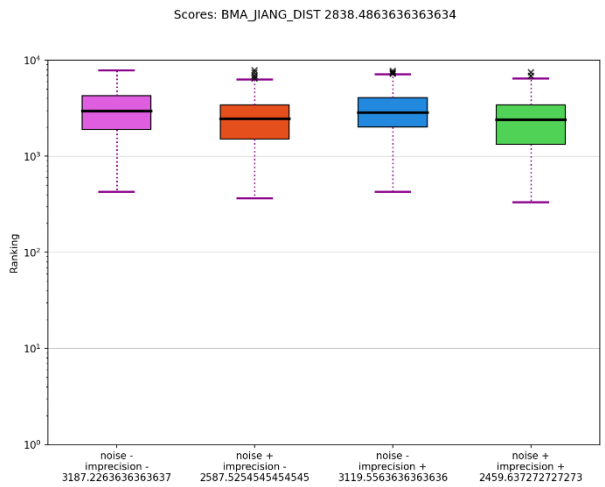
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	750	1.0
Noise + Imprecision -	638	1.0
Noise - Imprecision +	98	19.0
Noise + Imprecision +	73	28.5

Pairwise: Jiang and Conrath 1997

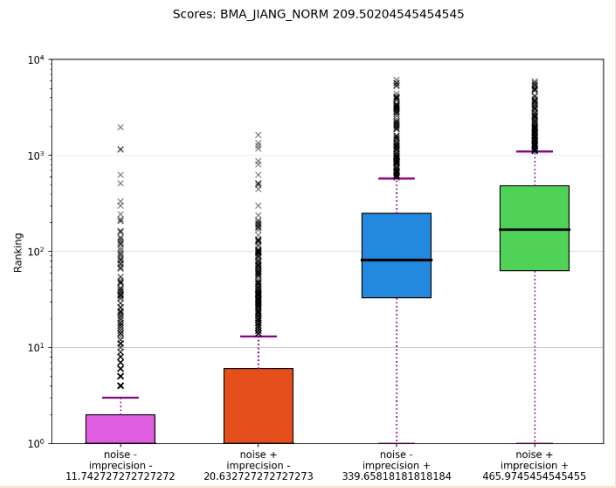
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	2970.0
Noise + Imprecision -	0	2454.5
Noise - Imprecision +	0	2871.5
Noise + Imprecision +	0	2405.0

Pairwise: Jiang and Conrath 1997 Normalized

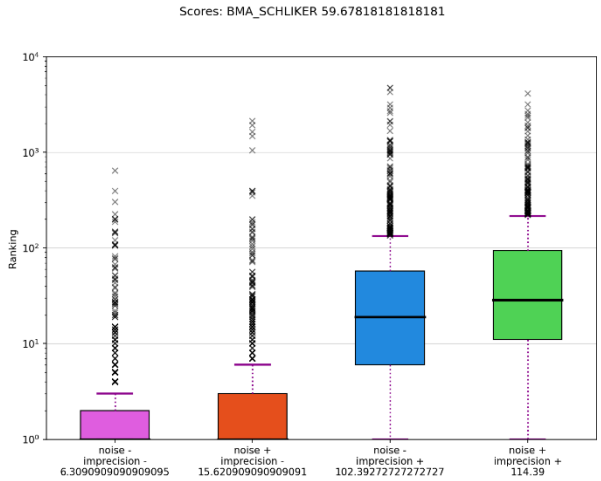
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	739	1.0
Noise + Imprecision -	584	1.0
Noise - Imprecision +	9	81.5
Noise + Imprecision +	2	169.0

Pairwise: Schlicker SimRel 2006

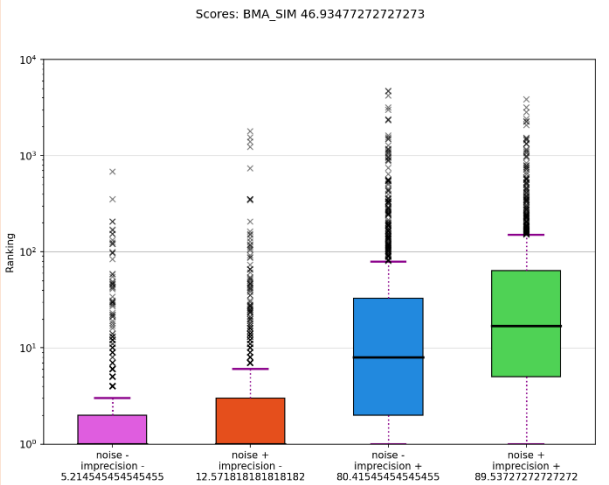
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	750	1.0
Noise + Imprecision -	638	1.0
Noise - Imprecision +	98	19.0
Noise + Imprecision +	73	28.5

Pairwise: SimIC 2010

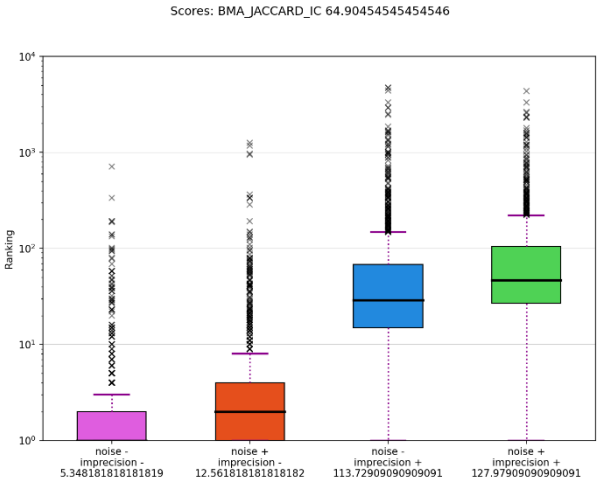
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	798	1.0
Noise + Imprecision -	650	1.0
Noise - Imprecision +	197	8.0
Noise + Imprecision +	74	17.0

Pairwise: Jaccard IC

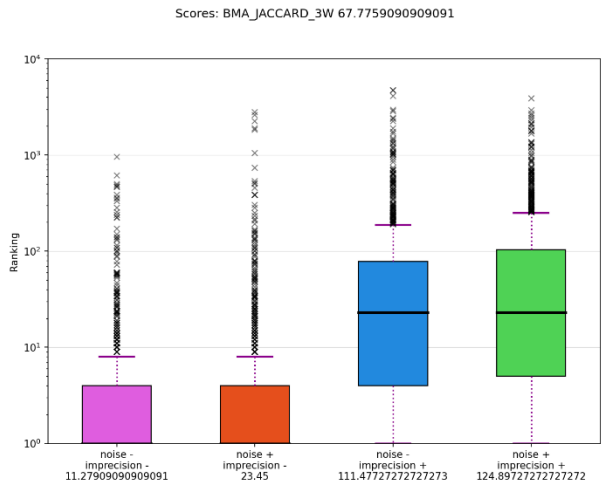
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	799	1.0
Noise + Imprecision -	540	2.0
Noise - Imprecision +	32	29.0
Noise + Imprecision +	5	47.0

Pairwise: Jaccard 3W IC

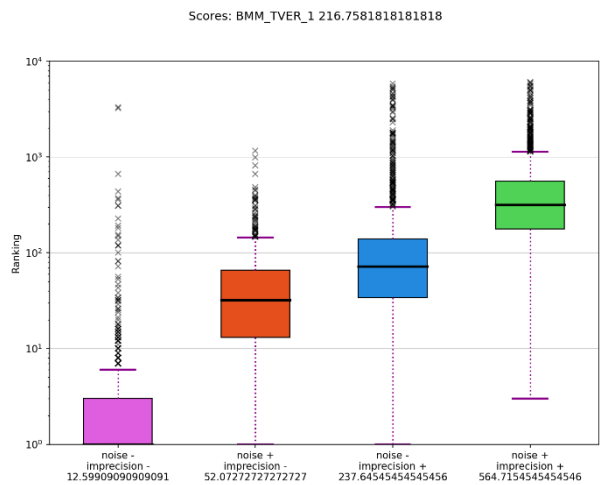
Indirect measure: Best Match Average (BMA)



	Nº of 1st ranks	Medians
Noise - Imprecision -	641	1.0
Noise + Imprecision -	643	1.0
Noise - Imprecision +	129	23.0
Noise + Imprecision +	133	23.0

Pairwise: Tversky IC Contrast model

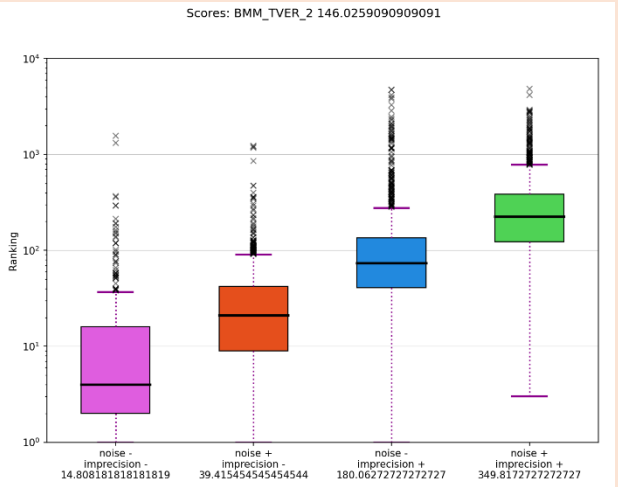
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians
Noise - Imprecision -	595	1.0
Noise + Imprecision -	36	32.0
Noise - Imprecision +	18	72.0
Noise + Imprecision +	0	318.0

Pairwise: Tversky IC Ratio model

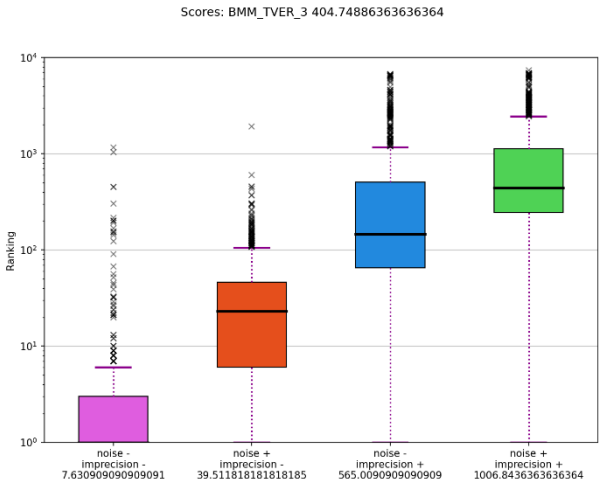
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians
Noise - Imprecision -	225	4.0
Noise + Imprecision -	31	21.0
Noise - Imprecision +	2	74.0
Noise + Imprecision +	0	224.0

Pairwise: Tversky IC Prop Contrast model

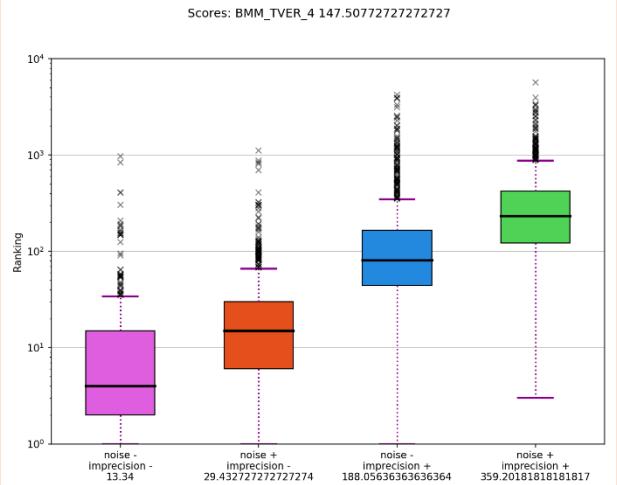
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians
Noise - Imprecision -	557	1.0
Noise + Imprecision -	59	23.0
Noise - Imprecision +	13	146.0
Noise + Imprecision +	2	441.5

Pairwise: Tversky IC Prop Ratio model

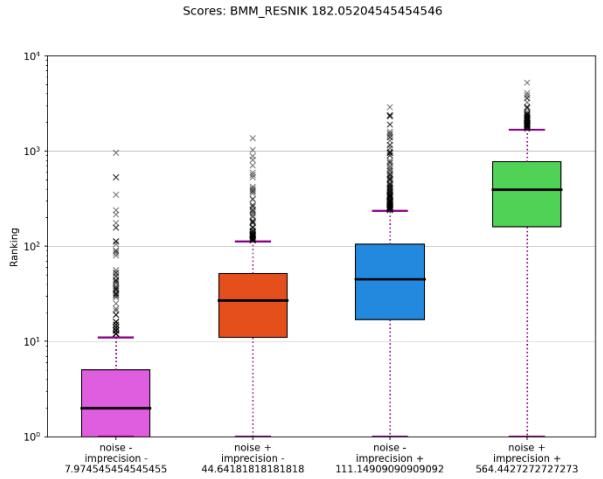
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians
Noise - Imprecision -	234	4.0
Noise + Imprecision -	53	15.0
Noise - Imprecision +	3	81.0
Noise + Imprecision +	0	233.5

Pairwise: Resnik 1995

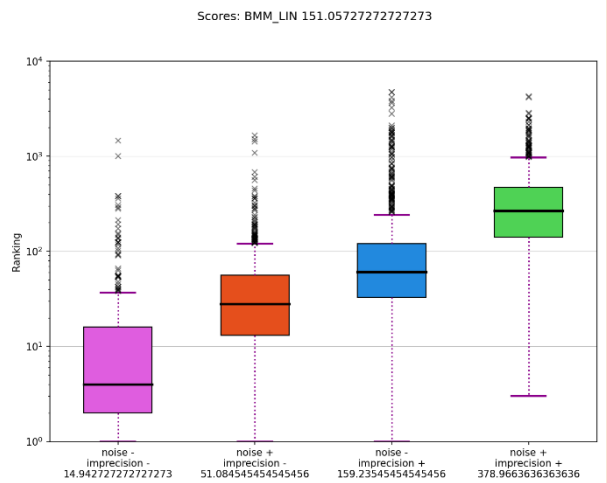
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians
Noise - Imprecision -	366	2.0
Noise + Imprecision -	47	27.0
Noise - Imprecision +	20	45.0
Noise + Imprecision +	1	394.0

Pairwise: Lin 1998

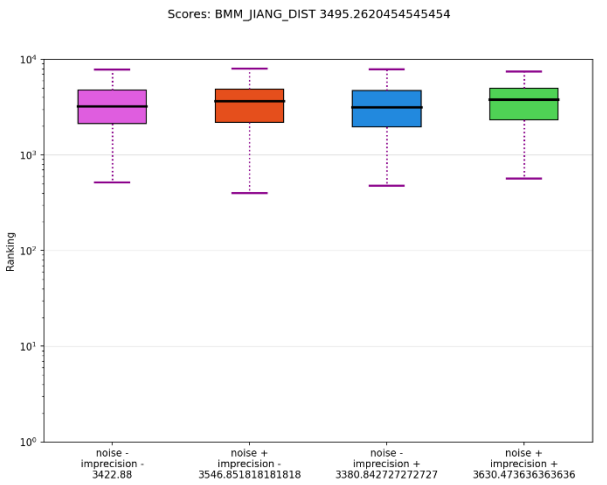
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians
Noise - Imprecision -	225	4.0
Noise + Imprecision -	24	28.0
Noise - Imprecision +	5	61.0
Noise + Imprecision +	0	268.0

Pairwise: Jiang and Conrath 1997

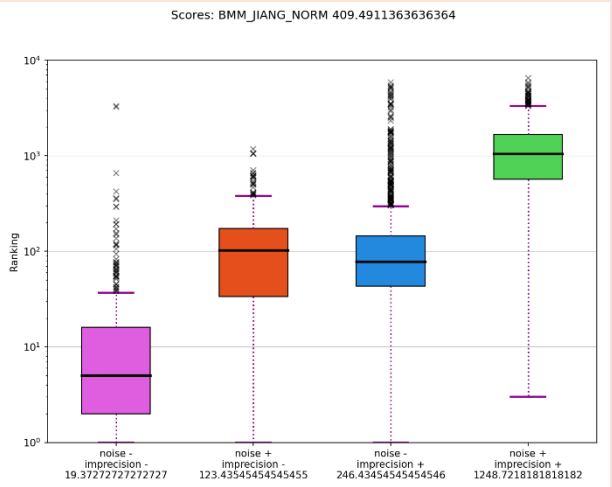
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	3242.0
Noise + Imprecision -	0	3676.0
Noise - Imprecision +	0	3166.5
Noise + Imprecision +	0	3809.0

Pairwise: Jiang and Conrath 1997 Normalized

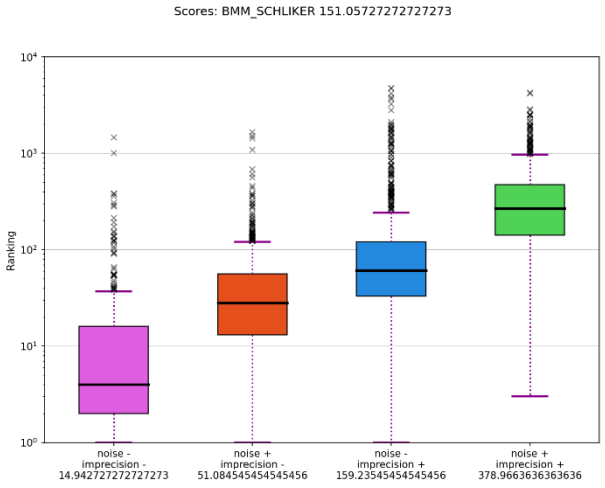
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians
Noise - Imprecision -	224	5.0
Noise + Imprecision -	29	103.0
Noise - Imprecision +	2	78.0
Noise + Imprecision +	0	1051.5

Pairwise: Schlicker SimRel 2006

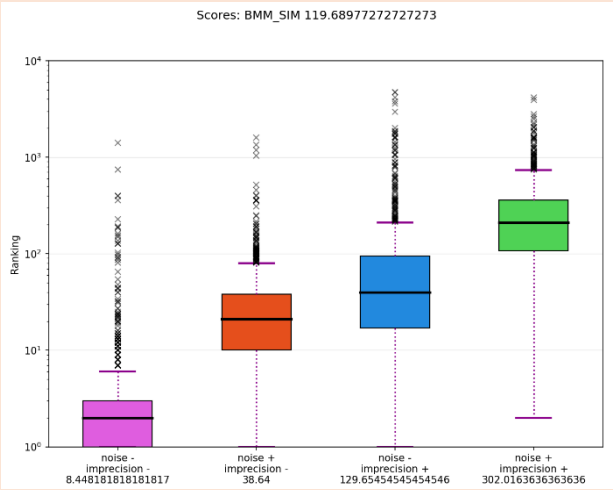
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians
Noise - Imprecision -	225	4.0
Noise + Imprecision -	24	28.0
Noise - Imprecision +	5	61.0
Noise + Imprecision +	0	268.0

Pairwise: SimIC 2010

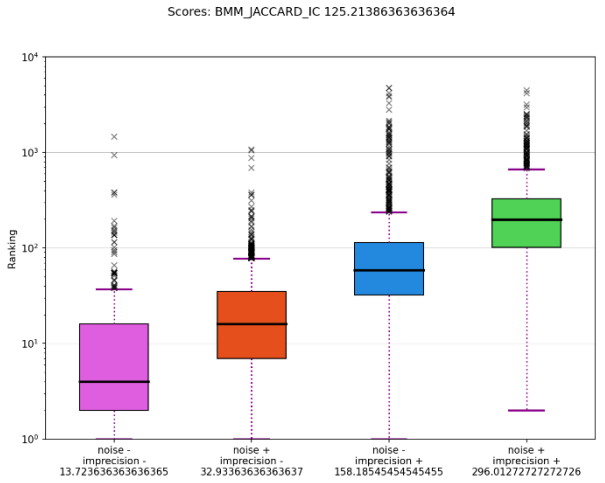
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians
Noise - Imprecision -	504	2.0
Noise + Imprecision -	36	21.0
Noise - Imprecision +	21	40.0
Noise + Imprecision +	0	210.0

Pairwise: Jaccard IC

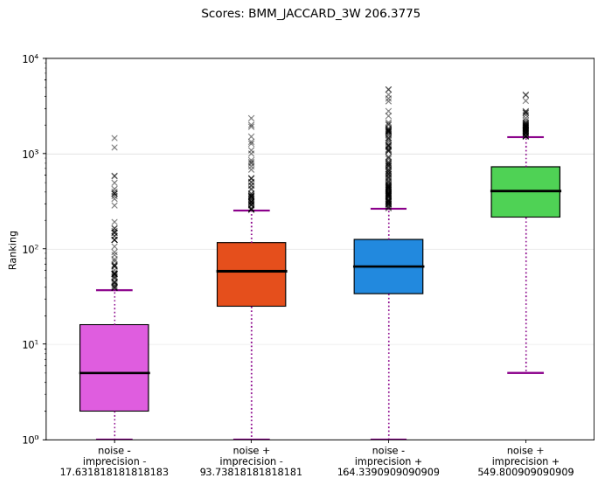
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians
Noise - Imprecision -	233	4.0
Noise + Imprecision -	48	16.0
Noise - Imprecision +	7	59.0
Noise + Imprecision +	0	198.5

Pairwise: Jaccard 3W IC

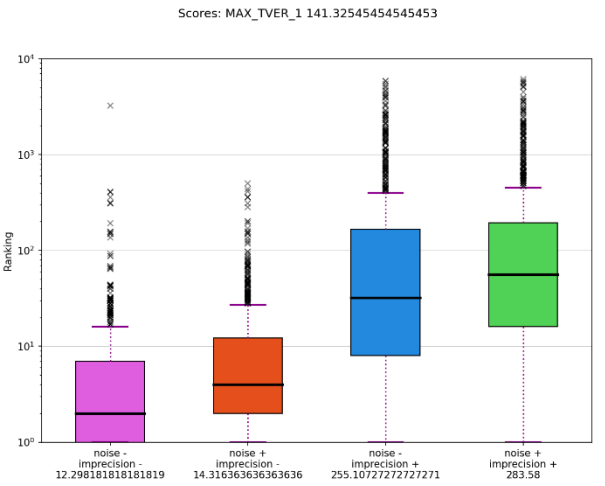
Indirect measure: Best Match Max (BMM)



	Nº of 1st ranks	Medians
Noise - Imprecision -	222	5.0
Noise + Imprecision -	16	59.0
Noise - Imprecision +	4	66.0
Noise + Imprecision +	0	408.5

Pairwise: Tversky IC Contrast model

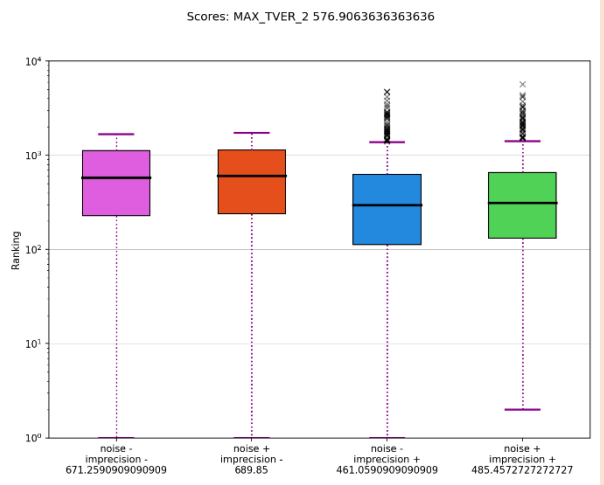
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	447	2.0
Noise + Imprecision -	272	4.0
Noise - Imprecision +	78	32.0
Noise + Imprecision +	51	56.0

Pairwise: Tversky IC Ratio model

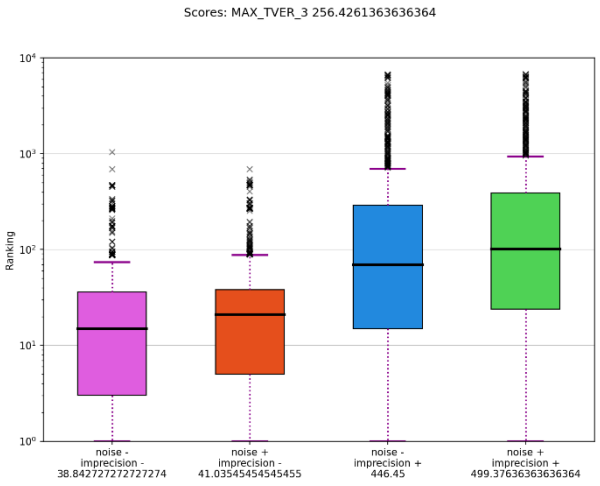
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	10	580.5
Noise + Imprecision -	8	608.0
Noise - Imprecision +	2	297.5
Noise + Imprecision +	0	313.5

Pairwise: Tversky IC Prop Contrast model

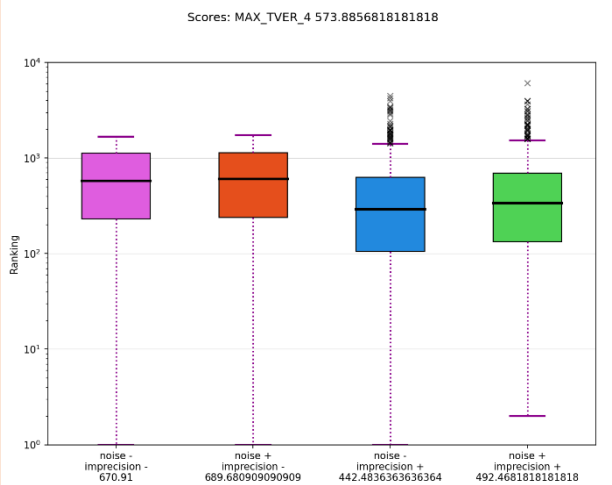
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	179	15.0
Noise + Imprecision -	133	21.0
Noise - Imprecision +	52	70.0
Noise + Imprecision +	35	101.0

Pairwise: Tversky IC Prop Ratio model

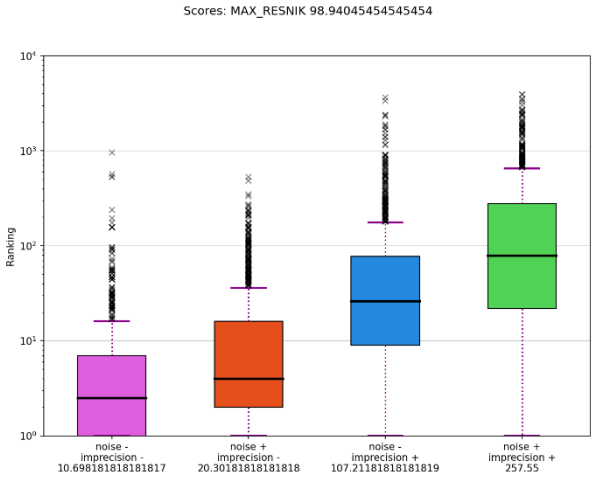
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	10	580.5
Noise + Imprecision -	8	608.5
Noise - Imprecision +	3	291.0
Noise + Imprecision +	0	338.0

Pairwise: Resnik 1995

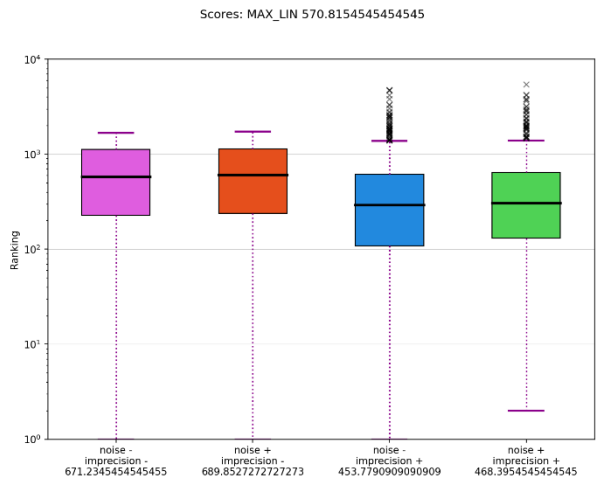
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	445	2.5
Noise + Imprecision -	265	4.0
Noise - Imprecision +	72	26.0
Noise + Imprecision +	50	79.0

Pairwise: Lin 1998

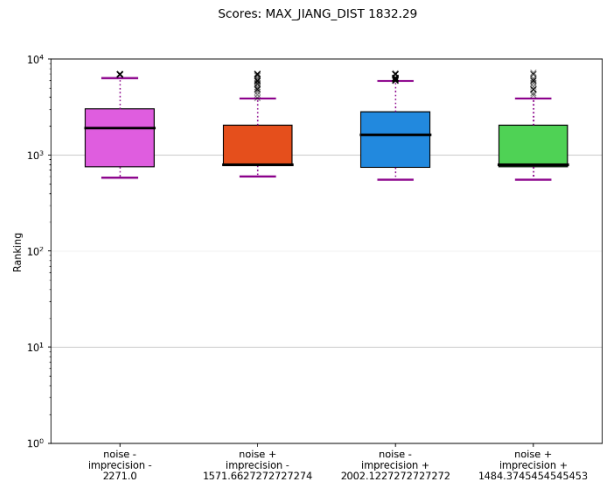
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	10	580.5
Noise + Imprecision -	8	608.0
Noise - Imprecision +	2	291.0
Noise + Imprecision +	0	308.0

Pairwise: Jiang and Conrath 1997

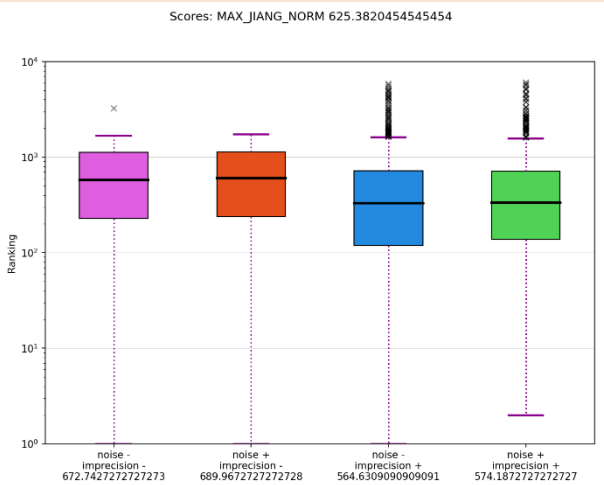
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	1926.5
Noise + Imprecision -	0	793.0
Noise - Imprecision +	0	1627.0
Noise + Imprecision +	0	793.0

Pairwise: Jiang and Conrath 1997 Normalized

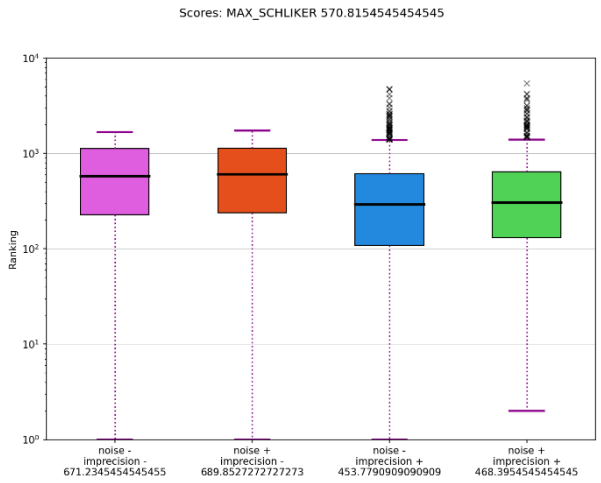
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	10	580.5
Noise + Imprecision -	8	609.5
Noise - Imprecision +	2	332.0
Noise + Imprecision +	0	334.0

Pairwise: Schlicker SimRel 2006

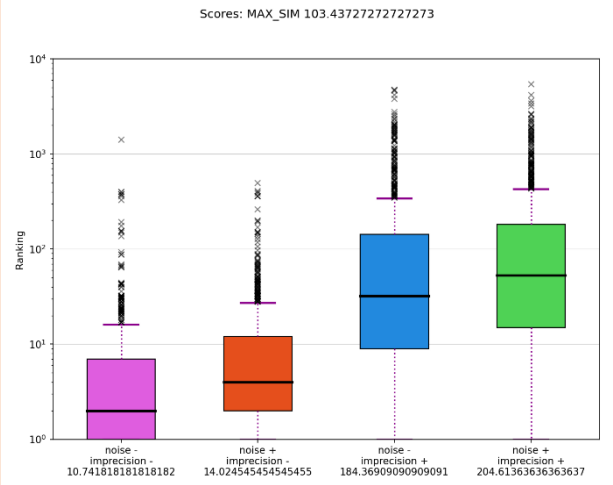
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	10	580.5
Noise + Imprecision -	8	608.0
Noise - Imprecision +	2	291.0
Noise + Imprecision +	0	308.0

Pairwise: SimIC 2010

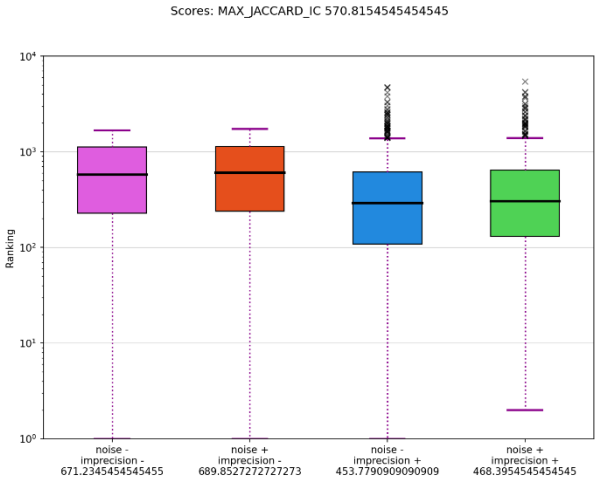
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	447	2.0
Noise + Imprecision -	272	4.0
Noise - Imprecision +	78	32.0
Noise + Imprecision +	51	53.0

Pairwise: Jaccard IC

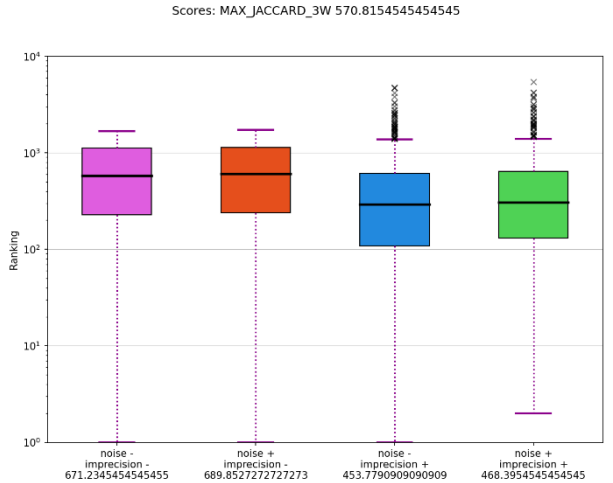
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	10	580.5
Noise + Imprecision -	8	608.0
Noise - Imprecision +	2	291.0
Noise + Imprecision +	0	308.0

Pairwise: Jaccard 3W IC

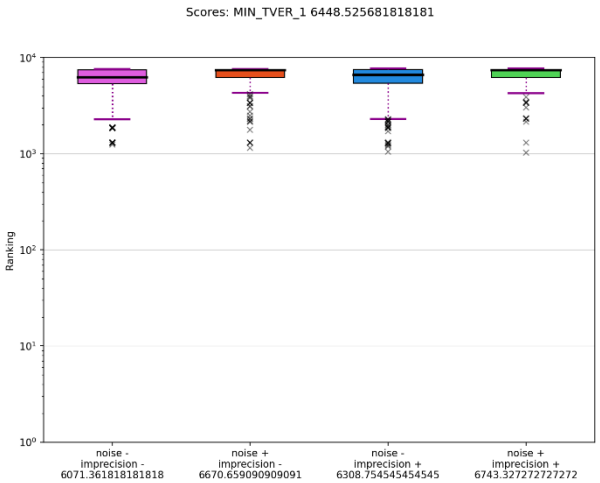
Indirect measure: Maximum



	Nº of 1st ranks	Medians
Noise - Imprecision -	10	580.5
Noise + Imprecision -	8	608.0
Noise - Imprecision +	2	291.0
Noise + Imprecision +	0	308.0

Pairwise: Tversky IC Contrast model

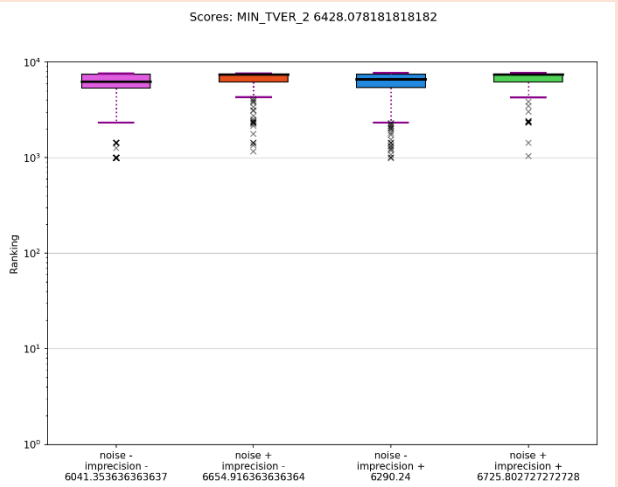
Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	6275.5
Noise + Imprecision -	0	7434.0
Noise - Imprecision +	0	6624.0
Noise + Imprecision +	0	7434.0

Pairwise: Tversky IC Ratio model

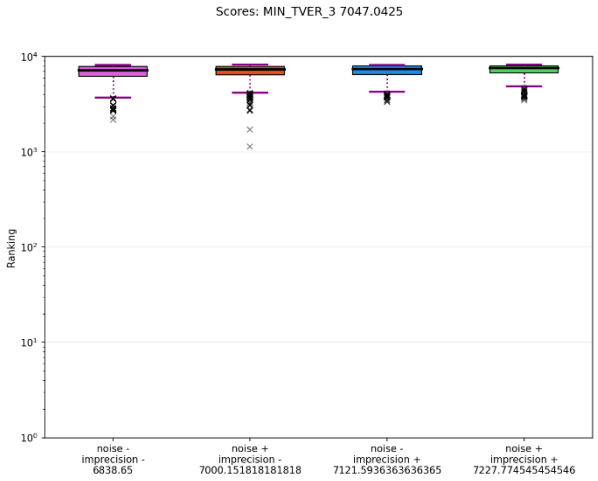
Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	6264.5
Noise + Imprecision -	0	7417.0
Noise - Imprecision +	0	6613.0
Noise + Imprecision +	0	7417.0

Pairwise: Tversky IC Prop Contrast model

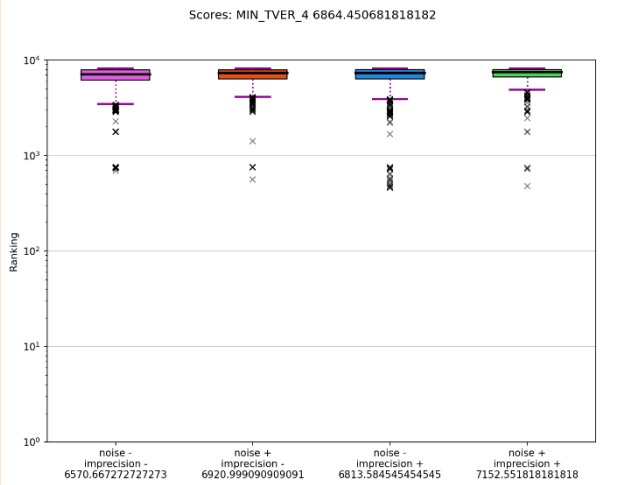
Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	7182.0
Noise + Imprecision -	0	7347.5
Noise - Imprecision +	0	7470.0
Noise + Imprecision +	0	7568.5

Pairwise: Tversky IC Prop Ratio model

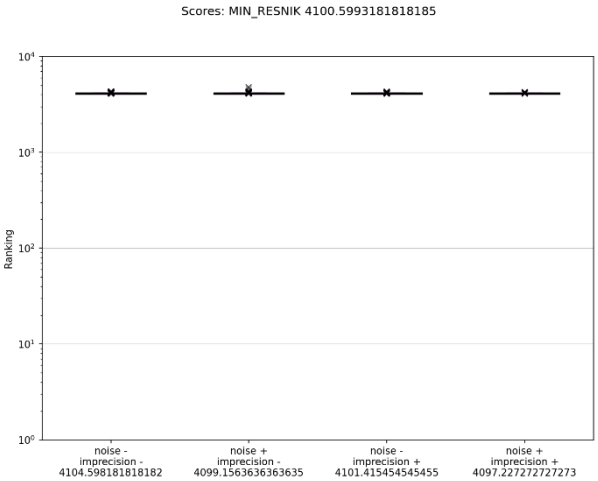
Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	7075.0
Noise + Imprecision -	0	7314.5
Noise - Imprecision +	0	7354.0
Noise + Imprecision +	0	7531.0

Pairwise: Resnik 1995

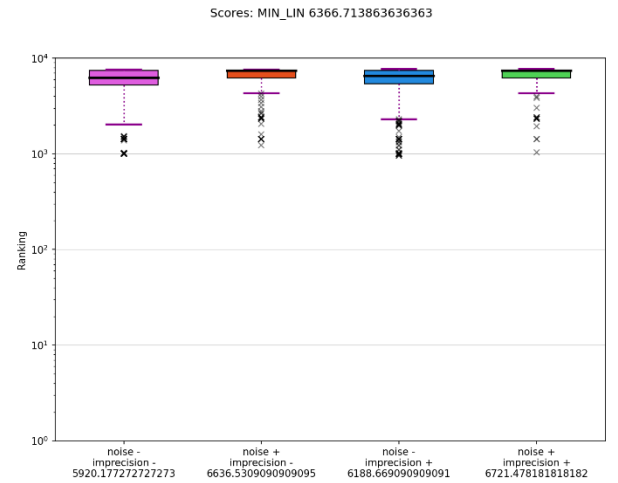
Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	4095.0
Noise + Imprecision -	0	4095.0
Noise - Imprecision +	0	4095.0
Noise + Imprecision +	0	4095.0

Pairwise: Lin 1998

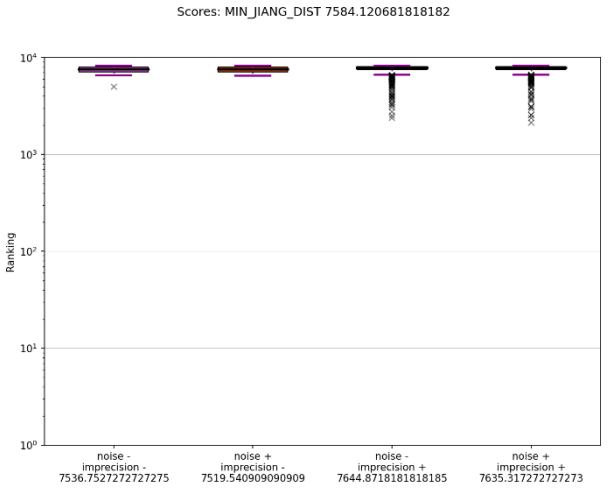
Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	6282.5
Noise + Imprecision -	0	7417.0
Noise - Imprecision +	0	6582.0
Noise + Imprecision +	0	7417.0

Pairwise: Jiang and Conrath 1997

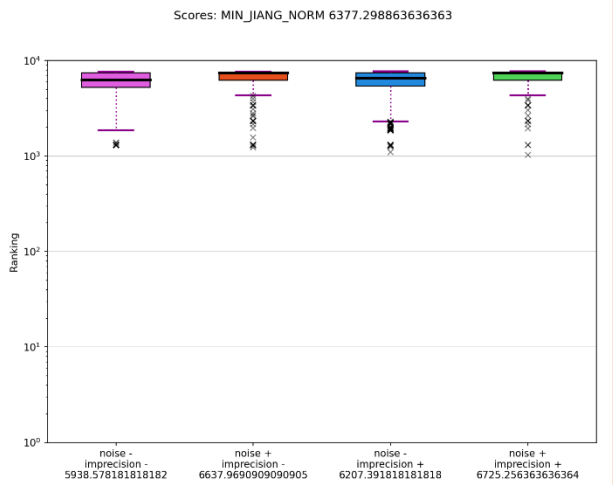
Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	7628.5
Noise + Imprecision -	0	7599.5
Noise - Imprecision +	0	7877.5
Noise + Imprecision +	0	7875.0

Pairwise: Jiang and Conrath 1997 Normalized

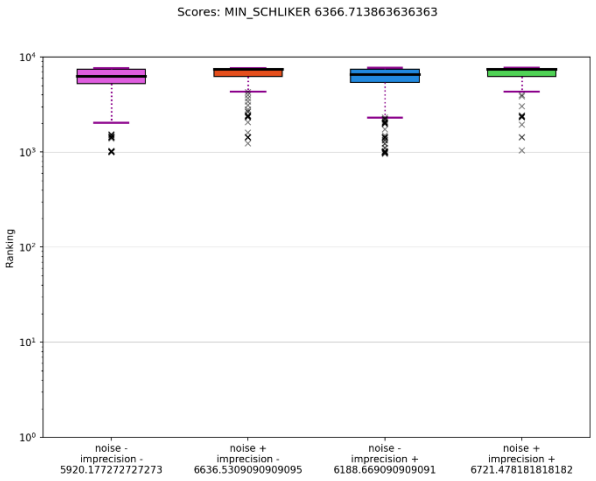
Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	6282.5
Noise + Imprecision -	0	7417.0
Noise - Imprecision +	0	6582.0
Noise + Imprecision +	0	7417.0

Pairwise: Schlicker SimRel 2006

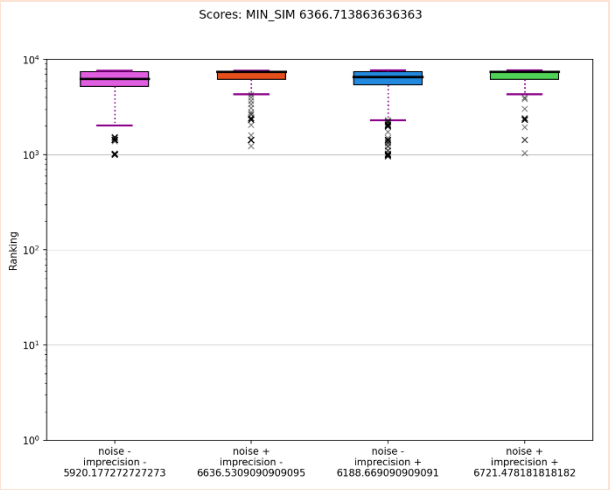
Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	6282.5
Noise + Imprecision -	0	7417.0
Noise - Imprecision +	0	6582.0
Noise + Imprecision +	0	7417.0

Pairwise: SimIC 2010

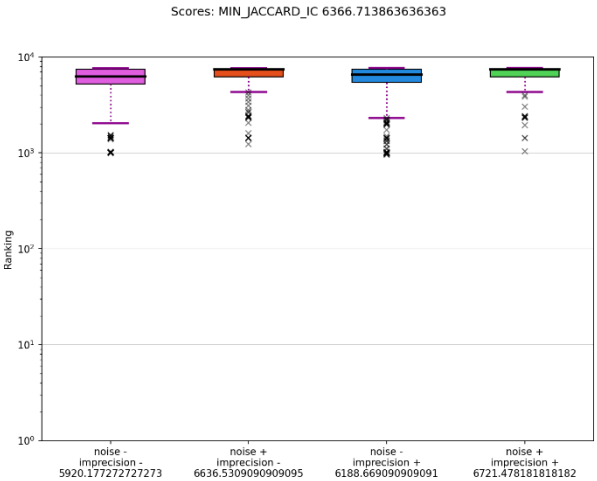
Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	6282.5
Noise + Imprecision -	0	7417.0
Noise - Imprecision +	0	6582.0
Noise + Imprecision +	0	7417.0

Pairwise: Jaccard IC

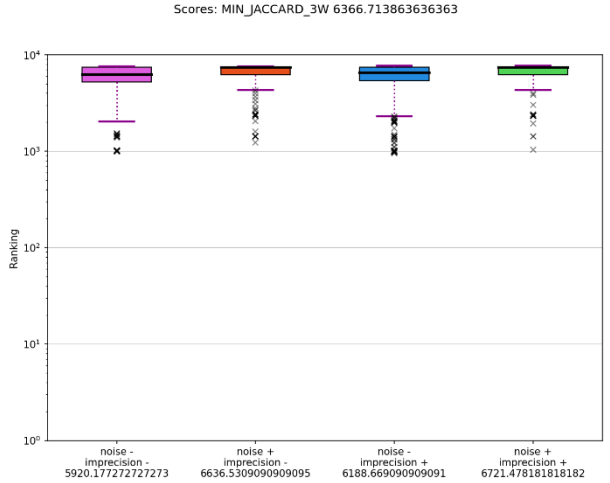
Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	6282.5
Noise + Imprecision -	0	7417.0
Noise - Imprecision +	0	6582.0
Noise + Imprecision +	0	7417.0

Pairwise: Jaccard 3W IC

Indirect measure: Minimum



	Nº of 1st ranks	Medians
Noise - Imprecision -	0	6282.5
Noise + Imprecision -	0	7417.0
Noise - Imprecision +	0	6582.0
Noise + Imprecision +	0	7417.0

Table 36. IC-based measures by similarity scores using the latest databases