Towards Efficient Indexing of Arbitrary Similarity

[Vision paper]

Tomáš Bartoš Tomáš Skopal Juraj Moško

Charles University in Prague, Faculty of Mathematics and Physics, SIRET Research Group
Malostranské nám. 25, 118 00 Prague, Czech Republic
{bartos, skopal, mosko}@ksi.mff.cuni.cz

ABSTRACT

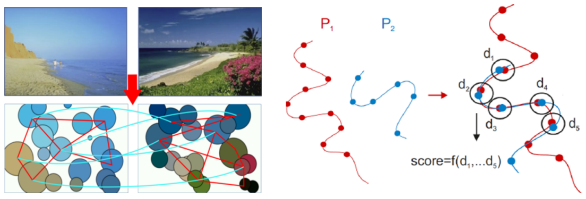
The popularity of similarity search expanded with the increased interest in multimedia databases, bioinformatics, or social networks, and with the growing number of users trying to find information in huge collections of unstructured data. During the exploration, the users handle database objects in different ways based on the utilized similarity models, ranging from simple to complex models. Efficient indexing techniques for similarity search are required especially for growing databases.

In this paper, we study implementation possibilities of the recently announced theoretical framework SIMDEX, the task of which is to algorithmically explore a given similarity space and find possibilities for efficient indexing. Instead of a fixed set of indexing properties, such as metric space axioms, SIMDEX aims to seek for alternative properties that are valid in a particular similarity model (database) and, at the same time, provide efficient indexing. In particular, we propose to implement the fundamental parts of SIMDEX by means of the genetic programming (GP) which we expect will provide high-quality resulting set of expressions (axioms) useful for indexing.

1. INTRODUCTION

The content-based retrieval is widely used in various areas of computer science including multimedia databases, data mining, time series, genomic data, social networks, medical or scientific databases, biometric systems, etc. In fact, searching collections of a priori unstructured data entities requires a kind of aggregation that ranks the data as more or less relevant to a query. A popular type of such a mechanism is the *similarity search* where, given a sample query object (e.g., an image), the database searches for the most similar objects (images). Two unstructured objects represented by their descriptors are compared by a similarity function, which produces a single numerical score interpreted as the degree of similarity between the two original objects.

For a long time, the database-oriented research



(a) Image similarity (b) Protein similarity

Figure 1: Sample similarity models

of similarity search employed the definition of similarity restricted to the *metric space* model with fixed properties of *identity*, *positivity*, *symmetry*, and especially *triangle inequality*, using *metric access methods* for indexing [2, 20, 14].

Together with the increasing complexity of data types across various domains, recently there appeared many similarities that were not metric – we call them *nonmetric* or *unconstrained* similarity functions [17]. As the nonmetric similarity functions are not constrained by any properties that need to be satisfied (unlike the metric ones), they allow to better model the desired concept of similarity and therefore lead to more precise retrieval (see Fig. 1a for a robust matching using local image features).

Also nonmetric similarities allow to design models that cannot be formalized into a closed-form equation. They could be defined as heuristic algorithms such as an alignment or a transformational procedure, while the enforcement of metric axioms could be very difficult or even impossible. As an example (see Fig. 1b), consider alignment algorithms for measuring functional similarity of protein sequences [18] or structures [8].

However, usually just the *database experts* are concerned with the existence of specific properties in a similarity function, as the properties enable the ways how to index the database for efficient similarity search. But database experts usually do not investigate the applicability of their techniques to specific domains. On the other hand, there are much

larger *domain expert* communities of different kinds – people who use specialized similarity search applications and are ready to apply any method in order to get expected results. These experts typically do not care about the indexing techniques or performance issues to a certain extent, so enforcement of any indexing-specific properties in their similarity functions is out of their expertise. For them, the best approach is to use the simplest (possibly inefficient) database methods as they are easy to implement. However, in long term and with large-scale databases, the efficiency will become a critical factor for choosing suitable similarity search methods.

Based on the different interests of database and domain research communities, the main goal of our research is to find a complex solution that provides the various domain experts with a database technique that allows effective similarity search yet that does not require any database-specific intervention to the generally unconstrained similarity models. In the following text, we shortly summarize previous attempts to unconstrained (nonmetric) similarity search before we sketch the idea of how to apply genetic programming for this purpose.

2. MOTIVATION

It is not always easy for domain experts to invent a perfect similarity measure, mostly represented as a distance (dissimilarity) function δ, and use it efficiently for large-scale databases with no compromise. The general way how to efficiently search is to use the *lowerbounding* principle – instead of computing expensive distances between a query object and all database objects a cheaper lowerbounding function LB is applied to filter the irrelevant ones.

The first lowerbounding approach might be to meet requirements of the *metric space model* by modifying the similarity model. Then a lowerbound function $LB_\triangle$ utilizing the *triangle inequality* is used

$$\delta(q,o) \geq LB_\triangle(\delta(q,o)) = |\delta(q,p) - \delta(p,o)| \quad (1)$$

for query q, pivot (reference) object p, and database object o. However, such a transformation might spoil the benefits of the original model.

So, the next option is to use an indirect variation of the model leveraging the known mapping approaches such as TriGen [15] which "converts" the nonmetric similarities into metric ones and, again, the metric model might be used. However, this is not always the best-case scenario as it might lead to either large retrieval error or low indexability [17].

Hence, there appeared some alternative methods of database indexing for unstructured data, such as the *Ptolemaic Indexing* [9, 11]. Here, the *Ptolemy's inequality* is used to construct lowerbounds. It states that for any quadrilateral, the pairwise products of opposing sides sum to more than the product of the diagonals. So, for any four database objects x, y, u, $v \in \mathcal{D}$, we have:

$$\delta(x,v)\cdot\delta(y,u) \leq \delta(x,y)\cdot\delta(u,v)+\delta(x,u)\cdot\delta(y,v) \quad (2)$$

For Ptolemaic lowerbounding LB_{ptol} with a given set of pivots $\mathbb{P}$, the bound δ_C derived from (2) is maximized over all pairs of distinct pivots [9, 11]:

$$\delta(q,o) \geq LB_{\text{ptol}}(\delta(q,o)) = \max_{p,s\,\in\,\mathbb{P}} \delta_{\text{C}}(q,o,p,s) \quad (3)$$

The ptolemaic indexing was successfully used with the *signature quadratic form distance* [11] that is suitable for effective matching of image signatures [1]. The idea of ptolemaic indexing shows that finding new indexing axioms could be a solution to speed-up similarity search in other way than mapping the problem to the metric space model.

3. RELATED WORK

We acknowledge that "lowerbounding problem" has been studied widely from various perspectives but as we found out this is true mostly for specific domains such as text or information retrieval (IR). For example, the recent paper [4] discusses axioms or constraints useful for term-weighting functions but it is limited to IR, while in [12] authors try to overcome improper lowerbounds with a new sufficiently large lowerbound for term frequency normalization (hardly applicable outside IR area).

Another work [13] reveals dynamic pruning strategies based on upper bounds to quickly determine the dissimilarity between an object and a query and thus quickly filter out objects; again designed for IR domain only.

Next, the definitions of axioms and constraints for similarity functions used in text retrieval systems are studied in [7], but the author provides only the theoretical background.

Interestingly, there exists a framework that provides an axiomatic approach for developing retrieval models [6]. It searches the spaces of candidate retrieval functions with the aim of finding the one that satisfies specific constraints. Although our approach might look the same, there are significant differences from our work. Particularly because authors are strongly connected to IR as they assume "bag-of-terms" representation of objects and they create retrieval functions inductively with respect to specific retrieval criteria. Most importantly, they focus on modeling the relevance rather than developing efficient database indexing techniques.

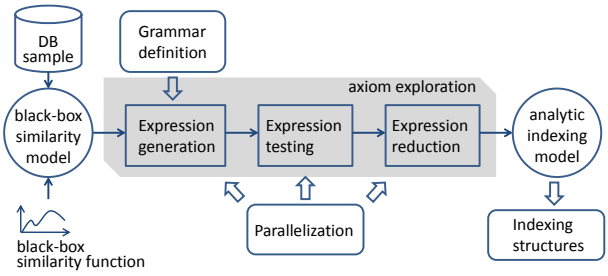


Figure 2: SIMDEX Framework high-level overview

So, a general method that provides a correct lowerbound for any domain has not been identified yet. And here we see the great potential for our research – to create and deliver a dataset-driven framework that is able to find lowerbounds for any given similarity space. This will then result in the efficient indexing method applicable to any domain.

4. SIMDEX FRAMEWORK

Our work outlines an alternative approach to similarity indexing motivated by the Ptolemaic indexing. Instead of "forcing" the distance and/or data to comply with the metric space model, for some datasets it could be more advantageous to employ completely different indexing model that provides cheap construction of lowerbounds. We intend to replace expensive distance computations between all pairs of objects by a cheaper lowerbounding function that filters out the non-interesting objects.

Therefore our major research goal is to develop a robust algorithmic framework for dataset-driven automatic exploration of axiom spaces for efficient and effective similarity search at large scale. We already described the SIMDEX framework and sketched a high-level overview (see Fig. 2) of the framework's stages (the inner components) in [16]. In that preliminary study, we designed only the theoretical concept while in this work, we verified our thoughts and clarify our vision with future steps.

4.1 Concept of SIMDEX Framework

As the input we consider a distance matrix for a *database sample* (S) computed with a *black-box distance function* (δ). This matrix consists of a set of values obtained by computing pair-wise distances between objects in the sample – it is our "mining field". The resulting output is a set of expressions (so called *axioms*) valid in the given similarity space that might be used for effective similarity search.

Using the basic idea of iteratively constructing and testing the expressions against the distance matrix, we are able to algorithmically explore axiom spaces specified in a syntactic way. This approach does not use a single canonized form and a tuning parameter, as other mapping approaches or the algorithm *TriGen* do. As the result, we will be able to discover the existing lowerbounding forms such as triangle inequality (Eq. 1) or Ptolemy's inequality (Eq. 3) as two instances in the axiom universe.

Moreover, since the resulting set of axioms (analytical properties) will be obtained in their lowerbounding forms, they can be immediately used for filtering purposes in the same way as ptolemaic indexing was implemented [11].

4.2 Framework Overview

In this section, we briefly introduce and describe the framework stages but for more details about particular components, we refer readers to our initial study in which the architecture and the methodology are described properly [16].

As the initial step, we use the grammar theory to create a *grammar definition* G based on which the expressions are subsequently generated. The generated expressions are in the standardized form of $\delta(q, o) \geq LB$, where LB will be expanded to various forms. Expressions cannot be computationally too expensive to evaluate and always include $\delta(\cdot, p)$, where pivot p is a fixed reference point.

Because the grammar-based generating of expression leads to an infinite universe, we limit the set of tested inequalities by (a) using the signatures of expressions that exclude various forms of the same expression (i.e., *fingerprints*), and (b) discarding meaningless expressions such as $\frac{x}{x}$, $-x$, ...

After we generate candidate expressions, they are tested against the precomputed distance matrix. As we require 100% precision, only such expressions are valid for which all tests are evaluated as `TRUE`.

To further condense the number of expressions we could refine the result by discarding weaker expressions or combining expressions into a compound expression, so only the best expressions will remain.

The last (indexing) step directly verifies the feasibility of the resulting set of expressions/axioms in practice within sample indexing tasks and validates the filtering power of each expression. We focus on the pivot table [2, 20] as it could be immediately used as an indexing structure for any kind of lowerbound expressions that involve pivots.

Although we optimize all stages, the exhaustive computation is still in place. Therefore, we assume massive parallelization of the exploration process leveraging classic multi-core CPU systems with multi-threading. For the future, we consider Map-Reduce technique [5] applied to a CPU farm or to a supercomputer architecture with lots of cores.

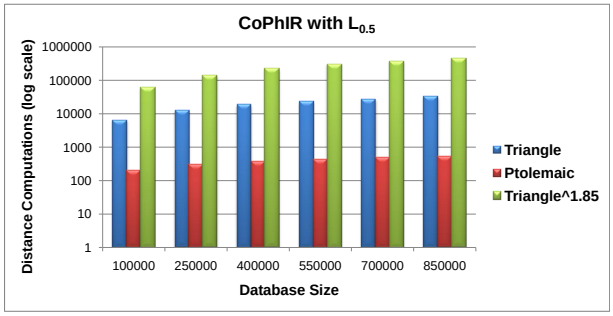


Figure 3: CoPhIR - Distance computations (log scale)

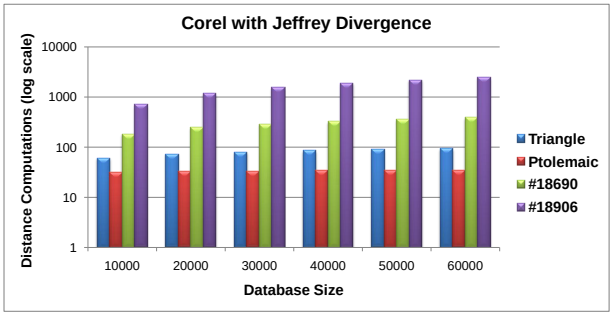


Figure 5: Corel - Distance computations (log scale)

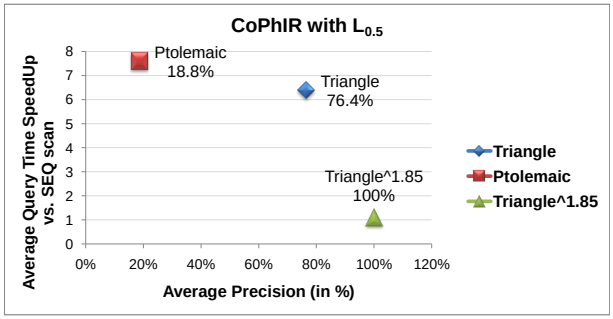


Figure 4: CoPhIR - Avg speedup vs. avg precision

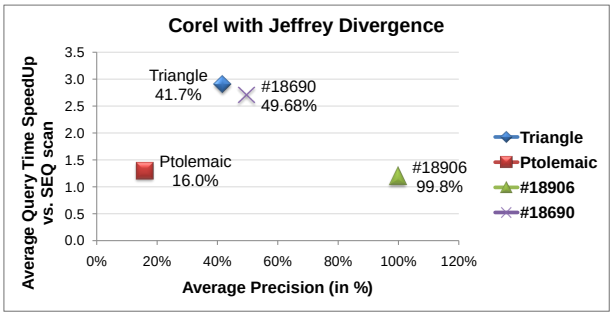


Figure 6: Corel - Avg speedup vs. avg precision

4.3 Preliminary results

After the naive implementation of all individual framework stages, we applied the prototype to the real-world datasets focusing on nonmetric similarity models in which metric postulates used for indexing and querying produced notable errors. This step validates our theoretical concept and as a proof we present convincing preliminary results.

Using a sample database (consisting of 25 objects), we tested CoPhIR[1] dataset with nonmetric $L_{0.5}$ distance and color histograms from *Corel Image Features*[2] dataset using nonmetric Jeffrey Divergence distance measure [17]. We verified the outcomes (resulting axioms) on indexing processes with Pivot Table [20] while studying the precision compared to results of sequential scan (SEQ), number of distance $\delta(\cdot,\cdot)$ computations (DCs) as the basic efficiency measure, and average speedup.

The best result for CoPhIR was the expression

$$\delta(q,o) \geq Triangle^{1.85}(\delta,q,p,o) = |\delta(q,p) - \delta(p,o)|^{1.85}$$

which does not dominate in number of DCs (Fig. 3) but it clearly produces no errors (Fig. 4) together with 1.1× speedup vs. SEQ scan.

[1]http://cophir.isti.cnr.it/
[2]http://goo.gl/SaOms

For Corel, we found the following expressions

$$\#18690 \quad \delta(q,o) \geq Triangle^{2}(\delta,q,p,o) = |\delta(q,p) - \delta(o,p)|^{2}$$
$$\#18906 \quad \delta(q,o) \geq (\delta(q,p_1) - \delta(o,p_1)) \cdot (\delta(q,p_2) - \delta(o,p_2))$$

While the squared triangle inequality (#18690) is only slightly more precise than triangle $LB_{\triangle}$ (Fig. 6), we achieved an enormous success with the next expression (#18906) – 99.8% precision together with 1.2× speedup compared to sequential scan. Although $LB_{\triangle}$ still dominates in the number of DCs (Fig. 5), it produces notable error rates (up to 59%).

4.4 Challenges

With the implemented prototype, we verified the feasibility of our concept; however, there appeared few issues that we need to overcome in order to provide a real and viable end-to-end solution. Namely, we need to address following challenges:

- **Expression Generation** – The basic concept of generating expressions iteratively covers all expressions (which is the advantage), however, a complex axiom valid in the given space could take enormous time to be revealed.
- **Expression Similarity** – Despite using the fingerprinting, we still struggle with testing only unique expressions and skipping the various forms of the similar ones, as there are infinite forms of how to express a single math expression.

- **Expression Testing** – We have to compromise between a large number of expressions to be tested and a bigger sample size. Testing the whole sample does not have to be always appropriate and we might take only some interesting objects from the sample.
- **Verifying indexing model** – To validate that resulting axioms could be used for indexing purposes, we run a separate indexing process on the data outside the sample which is correct but time-consuming.

5. GENETIC PROGRAMMING VISION

In order to improve and extend the framework capabilities and to overcome mentioned challenges (see Section 4.4), we propose using genetic programming (GP) as the main driver of generating and testing expressions. The concept of GP is not new and has been studied for several years since one of the first inspiring books was published [10]. In general, GP applies evolutionary patterns to a particular problem to achieve a specific goal using operations such as selection, crossover, or mutation [3].

We expect that GP-based approach will give the real power to the purely theoretical SIMDEX Framework (i.e., it will "materialize the theory"), will boost the efficiency of axiom discovery and speedup the axiom exploration process. Applying the principles of natural expression evolution will then lead to faster axiom resolution. Maybe we will not find all axioms valid in the given space but this is not our primary goal. In the first phase, we concentrate on detecting at least some axioms that will increase the efficiency of the indexing/filtering process.

5.1 GP-based SIMDEX Framework

Using GP-based method within the axiom exploration requires several customizations of individual framework stages. For this purpose, we propose and design the next generation of SIMDEX Framework (Fig. 7) which is how we perceive our future research. Connecting the existing theoretical concept together with GP-based algorithms (which will enrich it with the real and applicable context) we will gain a powerful tool for axiom exploration.

Our vision and the real motivator is, that given arbitrary user-defined similarity space, we will be able to find valid axioms within a *reasonable* and *acceptable* time frame. And we strongly believe GP-based components will help us to achieve this. Essentially, the novel GP-based axiom exploration process will address highlighted challenges with

- **Initial Population** - After we create the initial population with the existing expression

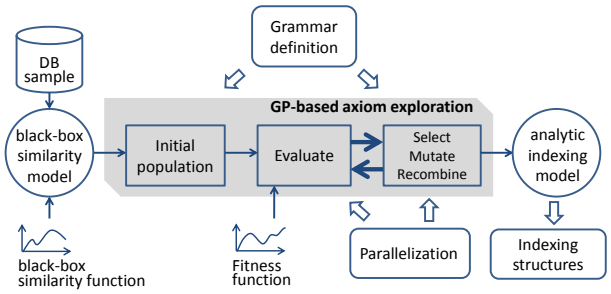


Figure 7: GP-based SIMDEX Framework

generator, additional expressions will be generated by the evolution algorithms which we expect will lead to "good" axioms early enough. We will consider two variants: *iteratively* and *randomly* built sets.

- **Evaluate** - This stage partially corresponds to Expression Testing, however we need to take into account several *fitness functions* to choose from such as (a) complete testing of a smaller distance matrix, (b) sampling n-tuples from a medium distance matrix, or (c) imitating a pivot-based search on a large distance matrix, which will give us better scalability of results.
- **GP-based operations** (Select, Mutate, Recombine) - Based on the evaluation results, we will *select* the most promising expressions and add them to the next generation. Some of them will be modified (*mutated*) or *recombined* with others (i.e., the crossover of expression trees) in order to boost their efficiency and find better expressions. During this stage, we need to test expression similarities and for this purpose, we consider applying a similarity measure to find similarities in expression trees (e.g., tree edit distance [15]) together with our previously proposed fingerprinting method.

We see the great potential in creating multiple generations of expressions based on the feedback from the evaluation, so we can try to modify the expressions to improve their efficiency accordingly. Depending on results, we will handle the mutation and recombination processes either in a completely random way, or there will be some logic behind to improve specific parts of an expression (modifying specific nodes in the expression tree).

The availability of multiple fitness functions gives us the opportunity to study expressions' behavior in different testing environments and potentially to come up with special characteristics of expressions and their suitability for specific datasets.

Another advantage is that GP has been studied and applied widely to lots of different areas and

there exists multiple options of how to perform each operation – sampling, recombination, or mutation, in order to obtain the next generation [19]. Therefore we can pick the method that will be mostly related and suitable to mathematical expressions.

6. CONCLUSION AND FUTURE WORK

With the preliminary implementation of purely theoretical **SIMDEX** Framework, we are able to demonstrate how to deal with the efficiency of similarity search in nonmetric spaces in other way than forcing the domain experts to implant and use metric postulates in their similarity models. Based on the results, we conclude that our framework is capable of finding alternative ways of indexing that speed up high-precision similarity queries.

However, to achieve this within an acceptable time frame and to find interesting axioms, we need to optimize it dramatically. For this purpose, we push our framework towards evolutionary algorithms (e.g., genetic programming). Doing so, we expect to explore the search space of all possible expressions more effectively and to have good results quickly. This method could provide better outcomes in terms of query efficiency/effectiveness for complex nonmetric similarity models. In the metric spaces, our solution will just provide a solid alternative to qualitatively dominating state-of-the-art techniques.

7. ACKNOWLEDGMENTS

This research has been supported by Grant Agency of Charles University (GAUK) projects 567312 and 910913 and by Czech Science Foundation (GAČR) project 202/11/0968.

8. REFERENCES

[1] C. Beecks, M. S. Uysal, and T. Seidl. Signature quadratic form distance. In *Proc. ACM International Conference on Image and Video Retrieval*, pages 438–445, 2010.

[2] E. Chávez, G. Navarro, R. Baeza-Yates, and J. L. Marroquín. Searching in metric spaces. *ACM Comp. Surveys*, 33(3):273–321, 2001.

[3] N. L. Cramer. A representation for the adaptive generation of simple sequential programs. In *Proc. of the 1st Int. Conf. on Genetic Algorithms*, pages 183–187. L. Erlbaum Associates Inc., USA, 1985.

[4] R. Cummins and C. O'Riordan. An axiomatic comparison of learned term-weighting schemes in information retrieval: clarifications and extensions. *Artif. Intell. Rev.*, 28:51–68, 2007.

[5] J. Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters. In *Proc. of the 6th conf. on Symp. on Oper. Systems Design & Impl.*, USA, 2004.

[6] H. Fang and C. Zhai. An exploration of axiomatic approaches to information retrieval. In *SIGIR*, pages 480–487. ACM, 2005.

[7] R. K. France. Weights and Measures: an Axiomatic Approach to Similarity Computations. Technical report, 1995.

[8] J. Galgonek, D. Hoksza, and T. Skopal. SProt: sphere-based protein structure similarity algorithm. *Proteome Science*, 9:1–12, 2011.

[9] M. L. Hetland. Ptolemaic indexing. `arXiv:0911.4384 [cs.DS]`, 2009.

[10] J. R. Koza. *Genetic programming*. MIT Press, Cambridge, MA, USA, 1992.

[11] J. Lokoč, M. Hetland, T. Skopal, and C. Beecks. Ptolemaic indexing of the signature quadratic form distance. In *Similarity Search and Applications*, pages 9–16. ACM, 2011.

[12] Y. Lv and C. Zhai. Lower-bounding term frequency normalization. In *Proc. of the 20th ACM Int. Conf. on Information and knowledge management*, CIKM '11, pages 7–16, New York, NY, USA, 2011. ACM.

[13] C. Macdonald, N. Tonellotto, and I. Ounis. On upper bounds for dynamic pruning. In *Proc. of the 3rd Int. Conf. on Advances in information retrieval theory*, ICTIR'11, pages 313–317. Springer-Verlag, 2011.

[14] H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann Publishers Inc., USA, 2005.

[15] T. Skopal. Unified framework for fast exact and approximate search in dissimilarity spaces. *ACM Transactions on Database Systems*, 32(4):1–46, 2007.

[16] T. Skopal and T. Bartoš. Algorithmic Exploration of Axiom Spaces for Efficient Similarity Search at Large Scale. In *Similarity Search and Applications*, LNCS, 7404, pages 40–53. Springer, 2012.

[17] T. Skopal and B. Bustos. On nonmetric similarity search problems in complex domains. *ACM Comp. Surv.*, 43:1–50, 2011.

[18] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147:195–197, 1981.

[19] D. Whitley. A genetic algorithm tutorial. *Statistics and computing*, 4(2):65–85, 1994.

[20] P. Zezula, G. Amato, V. Dohnal, and M. Batko. *Similarity Search: The Metric Space Approach*. Advances in Database Systems. Springer-Verlag, USA, 2005.