Fast approximate Duplicate Detection for 2D-NMR Spectra

Björn Egert[1], Steffen Neumann[1], and Alexander Hinneburg[2]

[1] Leibniz Institute of Plant Biochemistry, Department of Stress and Developmental Biology, Germany, {begert,sneumann}@ipb-halle.de
[2] Institute of Computer Science, Martin-Luther-University of Halle-Wittenberg, Germany, {hinneburg}@informatik.uni-halle.de

Abstract. 2D-Nuclear magnetic resonance (NMR) spectroscopy is a powerful analytical method to elucidate the chemical structure of molecules. In contrast to 1D-NMR spectra, 2D-NMR spectra correlate the chemical shifts of ^{1}H and ^{13}C simultaneously. To curate or merge large spectra libraries a robust (and fast) duplicate detection is needed. We propose a definition of duplicates with the desired robustness properties mandatory for 2D-NMR experiments. A major gain in runtime performance wrt. previously proposed heuristics is achieved by mapping the spectra to simple discrete objects. We propose several appropriate data transformations for this task. In order to compensate for slight variations of the mapped spectra, we use appropriate hashing functions according to the locality sensitive hashing scheme, and identify duplicates by hash-collisions.

1 Motivation

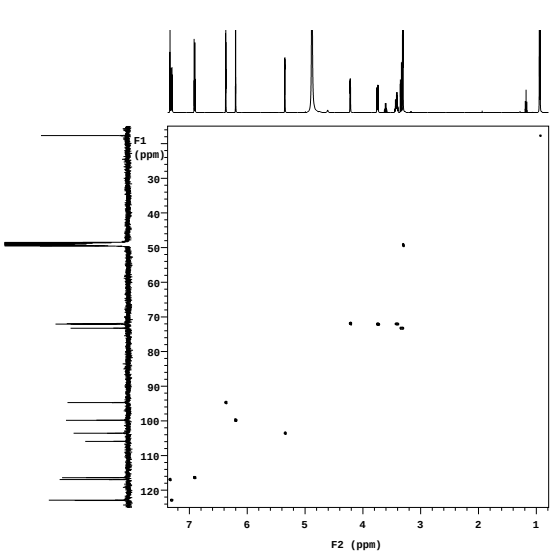
Nuclear magnetic resonance (NMR) spectra are important to analyze unknown natural products. In contrast to standard one-dimensional NMR spectroscopy, advanced two-dimensional NMR spectroscopy is able to capture the influences of two different atom types at the same time, e.g. ^{1}H (hydrogen) and ^{13}C (carbon).

The result of a 2D-NMR measurement can be seen as an intensity function measured over two independent variables[3]. Regions of the plane with high intensity are called peaks, which contain the real information about the underlying molecular structure. The usual visualizations of 2D-NMR spectra are contour plots as shown in figure 1 (^{1}H,^{13}C-HSQC NMR spectrum). [4] Contour lines in low intensity regions are clipped away, because they are produced by irreproducable fluctuations. An ideal peak would register as small dot. In the biochemical literature, peaks are noted by their two-dimensional positions.

However, due to the limited resolution available (depending on the strength of the magnetic field) multiple peaks may appear as a single merged object with non-convex shape, and after thresholding two different peaks, which are close

[3] The measurements are in parts per million (ppm).
[4] HSQC: Heteronuclear Single Quantum Coherence

together, may be merged and so both are represented by a single point. This is usually accepted. The pattern of peaks is very characteristic and specific for a particular substance.



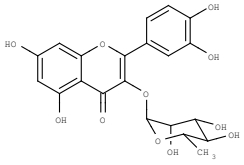


Fig. 1. 2D-NMR (HSQC) spectrum of Quercetrin, the one-dimensional plots at the axes are projections of the original two-dimensional intensity function including the respective signal intensities. Each peak captures characteristic ^{13}C,^{1}H- atomic resonance interactions present in the specific molecule.

As modern NMR devices allow the automatic analysis of many samples per day, the number of a spectra in a database can be up to several thousands per laboratory. Yet, manual work is needed to deduce the chemical structure of a complex organic substance from the spectrum. Thus, most of the NMR data is unpublished but contains a lot of experimental knowledge. Duplicate detection is needed for a use case where two or more libraries are merged, and the experimental knowledge for a pair of duplicates needs to be manually merged and curated. The matching has to be robust against merged peaks and measurements deviations between the two laboratories.

The problem is, given an automatically measured spectrum find all matching spectra on the basis of their peaks with annotations. We cast the specific problem in a more general setting: given a set of spectra find all pairs which are near-duplicates.

Our approach is based on a similarity measure with the desired robustness properties. In [15], we describe heuristics which guarantee no false negatives and reduce the average run time. However, the runtime complexities of those heuristics are still quadratic and the run times for very large data sets are still unacceptable.

In this paper, we propose to map the spectra to simple discrete objects like fixed length integer vectors or discrete sets, for which duplicates can be found much easier. The mapping may cause false negatives, as duplicate spectra may be

mapped to discrete objects with slight variations. The effect is compensated by searching similar discrete objects instead of identical ones. We use 1) manhattan distance and 2) the Jaccard coefficient for this task. For both similarity measures exist instances of the locality sensitive hashing scheme (LSH) [16], which uses a proper set of hashing functions to identify duplicate spectra by hash-collisions. The effectiveness of the proposed transformations are evaluated on real data with respect to quality and run time.

The remainder of the paper is organized as follows: after a discussion of related work in the next section, we introduce a simple definition of similarity and define fuzzy duplicates in section 3. Based on the exact method we discuss the transformation of spectra into discrete space in section 4, followed by the application of LSH to the problem. Our experiments are based on real data, their setup and results are shown in section 6. With the summary in section 7 we conclude the paper.

2 Related Work

Duplicate detection can be seen as a special case of content-based similarity search, where pairs of spectra are considered duplicates if their similarity exceeds a certain cutoff value. While content-based similarity search is already in use for 1D-NMR spectra [1, 2, 18, 19, 22], to the best of our knowledge, no effective similarity search method is known for 2D-NMR-spectra. Besides technical details (like how to choose the particular cutoff values for similarity) the problem of an approach purely based on similarity is, that the similarities between all pairs of spectra have to be computed. This leads to quadratic run time in the number of spectra, which is prohibitive for large spectra databases. In case of duplicate detection, more efficient algorithms exist.

Various aspects of detecting duplicates have received a lot of attention in database and information retrieval research. The closest type of approaches is near-duplicate detection of documents. The efficient detection of near-duplicate documents has been studied by several authors [5,24]. In particular, near-duplicate detection of web documents is a quite active research area [8, 12, 13]. The difference between near-duplicate documents and fuzzy duplicates of 2D-NMR spectra is that documents are composed of discrete entities, namely words or index terms, but 2D-NMR spectra consists of continuous 2D points. The crucial difference is that the matching operation is transitive for words but not for 2D points. An extension of near-duplicate documents are duplicates in XML documents [23], where the set of terms is organized as tree.

Duplicates are often found by using a similarity measure. Such measures can be manually defined, but in case of strings suitable similarity measures can be learned automatically using a support vector machine [3], which improves the detection accuracy. Another example of very difficult duplicates are those found in the WHO drug safety database [21]. In this case, a classification problem was solved in order to find a measure for comparison of the records. As those duplicates themselves are very difficult to detect, it seems unlikely to find

subquadratic algorithms for this problem class. Fortunately, fuzzy duplicates of 2D-NMR spectra have a more simple definition, which does not require advanced learning techniques.

The detection of duplicate records in data streams [9] or click streams [20] are new variants of the problem. Here, duplicates have simple definitions and the records have fixed length. NMR spectra have not that simple nature, e.g. the number of peaks may differ between spectra (due to the experimental setup even for chemical duplicates). Also the streaming scenario does not appear naturally for 2D-NMR spectra. However, the used technique, namely Bloom filters, are very promising and we will investigate in future research, whether Bloom filters can be applied in our scenario as well.

The detection of duplicates in images [17] is slightly related to our research, as 2D-NMR spectra could be thought as images as well. However, the used techniques in [17] ensure invariance wrt. scaling, shifting and rotation, which is not meaningful in case of 2D-NMR spectra.

The detection of duplicates is slightly related to collision detection in computer graphics [7]. The problem in this concern is to find 2D or 3D objects with overlapping boundaries in real time. The algorithms make the assumption, that only a few bounding boxes of the objects are overlapping. However, in our setting almost all bounding boxes of the spectra overlap. So, collision detection is not applicable to our problem.

Record linkage and especially the sorted neighborhood method [14] is also related to our approach. Sorted neighborhood determines for every object, in our case a 2D NMR spectrum, a key by which the objected are ordered. A sliding window is moved over the sorted sequence and objects within a window are checked for duplicates. The assumption behind the method is, that duplicates have the keys, which are close in the sorted object sequence. Key selection is crucial for the method. The sorted neighborhood method has been successfully used for identifying duplicates in customer databases with data objects consisting mainly of discrete attributes. Since those attributes ensure transitivity of duplicates, the key generation consists of selecting subsets of the discrete attributes. As 2D-NMR spectra do not have discrete attributes, the construction of a key is much more difficult. So far no promising technique is known for numeric attributes.

3 Definition of Similarity and Fuzzy Duplicates

A 2D-NMR spectrum of an organic compound captures characteristics of the chemical structure like rings and chains. As the shape of the measured peaks varies between experiments (even with the same substance!), we use centroid peak positions for the representation of the spectra. So, we define a spectrum as a set of two-dimensional points:

Definition 1. *A 2D-NMR spectrum A is defined as a set of points $\{x_1, \ldots, x_n\} \subset \mathbb{R}^2$. The $|\cdot|$ function denotes the size of the spectrum $|A| = n$.*

The number of peaks per spectrum is typically between 4 and 60. Our definition of duplicates is based on the idea that peaks can be matched. As spectra are measured experimentally, peak positions can differ even between technical replicates[5]. For that reason, peaks cannot be matched by their exact positions, but rather some slight deviations have to be allowed. A simple but effective approach is to match peaks only within a small spatial neighborhood, The neighborhood is defined by the ranges α and β:

Definition 2. *A peak x from spectrum A* ***matches*** *a peak y from spectrum B, iff $|x.c - y.c| < \alpha$ and $|x.h - y.h| < \beta$, where .c and .h denote the NMR measurements for carbon and hydrogen respectively.*

Based on the notion of matching peaks, we are ready to define a set-oriented similarity measure, from which in turn we derive the definition of duplicates as a special case. Note, that a single peak of a spectrum can match several peaks from another spectrum. Given two spectra A and B, the subset of peaks from A which find matching partners in B is denoted as $matches(A, B) = \{x\colon x \in A, \exists y \in B\colon x \text{ matches } y\}$. The function *matches* is not symmetric, but helps to define a symmetric similarity measure

Definition 3. *Let be A and B two spectra and $A' = matches(A, B)$ and $B' = matches(B, A)$, so* ***similarity*** *is defined as*

$$sim(A, B) = \frac{|A'| + |B'|}{|A| + |B|}$$

The measure is close to one if most peaks of both spectra are matching peaks. Otherwise, the similarity drops towards zero.

An important special case of similarity search is the detection of duplicates to increase the data quality of a collection of 2D-NMR-spectra. In addition to the measurement inaccuracies, in case a substance is measured twice with a high and low resolution, it may happen that neighboring peaks are merged to a single one. A restriction to one-to-one relationships between matching peaks can not handle such cases. This means that a single peak from spectrum A can be matching partner for two close peaks from spectrum B.

We propose a definition of fuzzy duplicates based on the similarity measure which can deal with the problems mentioned, namely deviances in peak measurements as well as splitted/merged peaks.

Definition 4. *A pair of 2D-NMR-spectra A and B are* ***fuzzy duplicates****, iff $sim(A, B) = 1$.*

By that definition it is only required that every peak of a spectrum finds at least one matching peak in the other spectrum. The parameters α and β can be set with the application knowledge of typical variances of single peak measurements. For our application, we chose $\alpha = 3$ ppm (^{13}C coordinate) and $\beta = 0.3$ ppm (^{1}H coordinate) if not stated otherwise.

[5] A technical replicate is the same substance/molecule under the same experimental conditions subjected to the measurement device at least twice.

3.1 Why is the problem difficult?

The duplicate definition is not transitive, that means if A is duplicate of B and B is duplicate of C that not necessarily A is duplicate of C. An example for this fact is sketched in figure 2. The reason is the nature of continuous measurements of the peak coordinates. The lack of transitivity has the consequence that a set

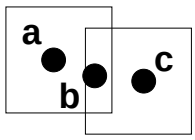


Fig. 2. The peak a from spectrum A matches peak b from spectrum B and b matches c from spectrum C. However a and c are not matching.

of duplicate spectra (where all spectra are pairwise duplicates) cannot be represented by a single spectrum. Such a representative would ease the detection of duplicates, since all duplicates of the representative are also pairwise duplicates. Because fuzzy duplicates of 2D-NMR spectra do not have this property, all pairs of the set have to be checked in order to calculate a set of duplicates. Thus, the complexity of an algorithm which finds all duplicates in a set of spectra has a quadratic worst case runtime $O(n)^2$ in the number of spectra n. Therefore, we have to resort to heuristics which reduce the experimental runtime on typical data sets.

4 Spectra Transformation

The exact methods [15], which are guaranteed to have no false negatives, do not scale to very large data sets, even when using peak selecting heuristics. Therefore, we investigate methods which have significantly lower run time. The price for the lower runtime is the possibility of false negatives, that means some duplicate pairs could be missed. We will discuss later how to avoid false negatives.

The problem of finding fuzzy duplicates of 2D-NMR spectra is, that the duplicate relation lacks transitivity. The reason is the continuous nature of the peak measurements. So, the idea is to map the peaks to some discrete objects. Among the many possibilities to do that, we will explore two principal alternatives of those mappings. First, the peak coordinates are discretized and then those integers are concatenated to a fixed length vector. Second, the peaks of a spectrum are mapped to discrete objects so that a spectrum is represented by a set of those objects.

The task of finding duplicate spectra is then reduced to finding duplicates of integer vectors and duplicate sets of discrete objects respectively. Both of the latter duplicate relations are transitive, so that a set of duplicates can be specified by a single representative vector or set. In order to check whether a new mapped spectrum belongs to a set of duplicates, it suffices to test the duplicate relation with the representative of the set.

False negatives occur in this approach, when duplicate spectra are mapped to different discrete objects. We propose mappings which map duplicate spectra to discrete objects which are – if not identical – at least very similar.

4.1 Mapping to Integer Vectors

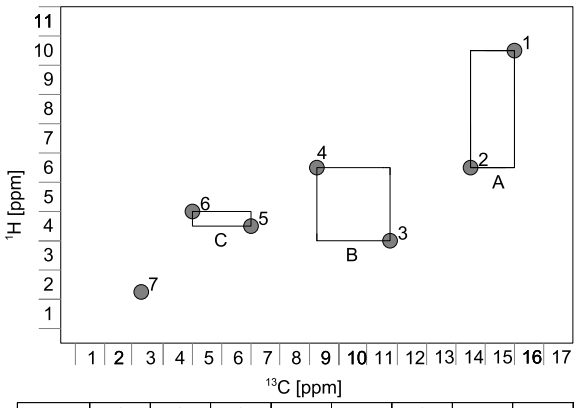
The first proposed mapping of 2D-NMR spectra maps transformed peaks to coordinates of the discrete integer vectors. Such a mapping involves three issues, namely (1) how to handle possible splits/merges of peaks, (2) how to order the transformed peaks to a vector, and (3) how to chose the overall dimensionality of the vectors.

Robustification: In order to handle the problem of peak splitting, some peak x of a spectrum is selected and those peaks y are deleted from the same spectrum which are in the neighborhood of x. The neighborhood is given by $N(x) = \{y\colon y \neq x, |x.c - y.c| \leq \alpha \text{ and } |x.c - y.c| \leq \beta\}$. The peaks are selected in decreasing order of $|N(x)|$, so that the peak with the largest number of neighbors is selected first. The iteration stops when each peak in the spectrum is a singleton, i.e. the neighborhoods of the remaining peaks are empty. The remaining peaks are called the *representative peak set* of a spectrum. After this step, a one to one relation between between peaks of duplicate spectra can be assumed.

Peak Ordering: The coordinates of the representative peaks of a spectrum are discretized by binning. The question remains how to order the discretized peak coordinates to form a vector, so that the order is not affected by small measurement errors. The most robust order is to sort ^{13}C- and ^{1}H-coordinates independently and discretize afterwards. The vector consists of a block of ^{13}C-coordinates followed by a block of ^{1}H-coordinates. However, this procedure would entirely ignore the joint distribution of ^{13}C- and ^{1}H-measurements but resorting to the marginal distributions only. So, quite different spectra could be mapped to the same integer vector.

The other extreme is to sort the peaks by one coordinate – say ^{13}C – only, and form a vector of alternating discretized ^{13}C- and ^{1}H-coordinates. The information of the joint distribution of ^{13}C- and ^{1}H- coordinates is retained in this mapping. In case of two peaks with close ^{13}C-coordinates but different ^{1}H-coordinates, measurement errors in the ^{13}C-coordinate of a duplicate spectrum could result in swaped order of the two peaks, which in effect also swaps the positions of the ^{1}H-coordinates. In case of two spectra being duplicates their integer vectors could be quite dissimilar, because of the difference in the swaped ^{1}H-coordinates.

We propose an intermediate approach, which combines the robustness of the first with the discrimination power of the second. The representative peaks of a spectrum are sorted by one coordinate, say ^{13}C. Starting with the peak of the largest ^{13}C-coordinate, we use a jumping window of w consecutive peaks. We sort the ^{13}C- and ^{1}H- coordinates independently for the w peaks inside a window, and arrange them in blocks as in the first approach. The last window might contain less than w peaks if $\#peaks \mod w \neq 0$. The important aspect of this technique is, that peaks in the close neighborhood from another spectrum



Peak	1		2		3		4		5		6		7			
Window	A				B				C				D			
Axis	C		H		C		H		C		H		C	H		
Bin#	15	14	10	6	11	9	6	4	6	4	5	4	3	2		

Fig. 3. Mapping of peaks from a spectrum to integer vectors for $w = 2$. The blocks of the peaks are indicated by rectangles. The resulting integer vector of the discretized spectrum is shown in the table underneath (last row). The windows and C and H blocks within a window are shown in the second and third row respectively.

map to the same sorted blocks, regardless of their order in the ^{13}C- axis. The problem of the second extreme approach can only occur at the jump positions. So, by choosing w we can search for a tradeoff between robustness and retained information. The process is illustrated in figure 4.1.

Although some peaks of duplicate spectra might map to different integer vectors due to the binning process, i.e. close peaks coordinates are mapped to different bins, the difference is at most one bin per coordinate.

Overall dimensionality: The overall dimensionality D of the set of resulting spectra vectors S is determined by the spectrum having the largest set of representative peaks $D = max(\#peaks(S_i))$. Since the spectra have different numbers of representative peaks, we need to pad their integer vectors up to the fixed dimensionality D. Padding the vectors with zeroes increases their overall similarity, whereas padding by random values would decrease their overall similarity. Therefore we pad a vector by repeating the vector itself until the the length of the maximal vector is reached, thereby retaining the similarity of the original vectors.

4.2 Mapping to Discrete Sets

We introduce a simple grid-based mapping to map a spectrum to a set of discrete objects, on which we will build a more sophisticated method.

Simple Grids A simple grid-based method is to partition each of the both axis of the two-dimensional peak space into intervals of same size. Thus, an equidistant grid is induced in the two-dimensional peak space and a peak is mapped to exactly one grid cell it belongs to. When a grid cell is identified by a

discrete integer vector consisting of the cells coordinates the mapping of a peak $x \in \mathbb{R}^2$ is formalized as

$$g(x) = (g_c(x.c), g_h(x.h)) \text{ with } g_c(x.c) = \left\lfloor \frac{x.c}{\alpha} \right\rfloor,\ g_h(x.h) = \left\lfloor \frac{x.h}{\beta} \right\rfloor$$

The quantities α and β are the extensions of a cell in the respective dimensions. The grid is centered at the origin of the peak space.

Shifted Grids A problem of the simple grid-based method is that peaks which are very close in the peak space may be mapped to different grid cells, because a cell border is between them. So proximity of peaks does not guaranty that they are mapped to the same discrete cell.

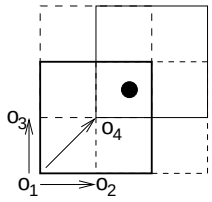


Fig. 4. The four grids are marked as follows: base grid is bold, $(1,0)$, $(0,1)$ are dashed and $(1,1)$ is normal.

Instead of mapping a peak to a single grid cell, we propose to map it to a set of overlapping grid cells. This is achieved by several shifted grids of the same granularity. In addition to the base grid some grids are shifted into the three directions $(1,0)(0,1)(1,1)$. An illustration of the idea is sketched in figure 4. In figure 4, one grid is shifted in each of the directions by half of the extent of a cell. In general, there may be $s-1$ grids shifted by fractions of $1/s, 2/s, \ldots, s-1/s$ of the extent of a cell in each direction respectively. For the mapping of the peaks to words which consist of cells from the different grids, two additional dimensions are needed to distinguish (a) the $s-1$ grids in each direction and (b) the directions themselves. The third coordinate represents the fraction by which a cell is shifted and the fourth one represents the directions by the following coding: value 0 is (0,0), 1 is (1,0), 2 is (0,1) and 3 is (1,1). So each peak is mapped to a finite set of four-dimensional integer vectors. A nice property of the mapping is that there exists at least one grid cell for every pair of matching peaks both peaks are mapped to.

5 Approximate Methods as Filter

The proposed mappings of the 2D-NMR data to discrete objects cannot guarantee, that duplicate spectra are mapped exactly to the same discrete objects. However, the mappings are designed in a way, that the mapped duplicate spectra are at least very similar discrete objects. In this section we focus on methods, which approximate similarity measures for those discrete objects (i.e. integer vectors and discrete sets).

5.1 Locality Sensitive Hashing

A general approximation scheme is locality sensitive hashing (LSH) [16], which is a distribution on a family of hash functions F on a collection of objects, such that for two objects x, y

$$Pr_{h \in F}[h(x) = h(y)] = sim(x, y)$$

The idea is to construct k hash functions h on the set of objects according to the family F. The percentage of collisions among the k pairs of hash values for two objects estimates the probability of a collision and gives an approximative similarity score. In general, the outcome of a hash function can be thought of as an integer. So, the LSH-scheme maps each object to a k-dimensional integer vector.

In case, two objects x, y are very similar, their integer vectors agree on all k coordinates with high probability. Let be $s = sim(x, y),\ s \in [0, 1]$ the similarity between x, y, then the probability is s^k that $h_i(x) = h_i(y)$ agree for all $1 \leq i \leq k$. To amplify that probability, the sampling process is repeated L times [10]. So, after L repetitions the probability that their integer vectors agree on all k coordinates at least once is

$$Pr[1 \leq i \leq k \colon h_i(x) = h_i(y) \text{ at least once}] = 1 - (1 - s^k)^L$$

Thus, the duplicate detection consist of finding L times the duplicates among integer vectors and union the results. Finding groups of equal integer vectors can be done by sorting, which has lower run time complexity than the naive algorithm.

There are locality sensitive hashing schemes known for the following similarity functions, Manhattan distance between fixed length integer vectors [11], and Jaccard coefficient for set similarity [4,6]. We briefly review the hashing schemes for the similarity measures.

5.2 Manhattan Distance

Given a set of d-dimensional integer vectors with coordinates in the set $\{1, \ldots, C\}$, the Manhattan distance between two vectors is $x, y \in X,\ d_1(x, y) = \sum_{i=1}^{d} |x_i - y_i|$. Let be $x = (x_1, \ldots, x_d)$ a vector from X and $u(x) = \text{Unary}_C(x_1) \ldots \text{Unary}_C(x_d)$ a transformation of x into a bit string, where $\text{Unary}_C(a)$ is the unary representation of a with C bits, i.e. a sequence of a ones followed by $C - a$ zeros. For any two vectors $x, y \in X$ there is $d_a(x, y) = d_H(u(x), u(y))$ with d_H is the Hamming distance, which gives the number of different bits between bit strings. An appropriate family of hash functions with the LSH property consists of $h_i(b)$, $1 \leq i \leq \text{length}(b)$, where $h_i(b)$ returns the ith bit from b.

Sampling uniformly from those hash functions and testing for collisions reduces to probabilistically counting the number of equal bits:

$$d_1(x, y) = d_H(u(x), u(y)) = dC(1 - Pr[h_i(u(x)) = h_i(u(y))])$$

with random h_i, $1 \leq i \leq dC$.

For the implementation of this LSH scheme, k random indices $i_1, \ldots i_k$ are picked. The transformation into the Hamming space, which can be quite large, is in practice not necessary. In order to find the value of $h_i(u(x))$ we have to look to which coordinate of the integer vector the index i belongs and if $(i-1 \mod C)+1$ is larger than the integer value of that coordinate. So the hash function for index i is

$$h_i(u(x)) = \begin{cases} 1 & \text{if } (i-1 \mod C)+1 \leq x_{\lfloor \frac{i}{C} \rfloor + 1} \\ 0 & \text{else} \end{cases}$$

5.3 Approximate Cosine Similarity

Cosine similarity is used in information retrieval to compare documents which are represented by term frequency vectors. Given a subset $A \subset U$ of a universe U the term frequency vector $\boldsymbol{t}_A$ has $|U|$ components, each representing the number of occurrences of a particular element in A. The cosine similarity of A, B is

$$sim_C(A,B) = \frac{\boldsymbol{t}_A \cdot \boldsymbol{t}_B}{\|\boldsymbol{t}_A\| \cdot \|\boldsymbol{t}_B\|}$$

The hash functions are constructed by randomly mapping each element of U to $\{-1,1\}$. Lets represent such a mapping $m\colon U \to \{-1,1\}^{|U|}$ as a vector $\boldsymbol{m}$, then the hash function induced by m is

$$h_{\boldsymbol{m}}(A) = \begin{cases} 1 & \text{if } \boldsymbol{m} \cdot \boldsymbol{t}_A \geq 0 \\ 0 & \text{if } \boldsymbol{m} \cdot \boldsymbol{t}_A < 0 \end{cases}$$

The LSH scheme is then

$$Pr[h_{\boldsymbol{m}}(A) = h_{\boldsymbol{m}}(B)] = 1 - \frac{\theta(\boldsymbol{t}_a, \boldsymbol{t}_b)}{\pi/2} \approx sim_c(A,B)$$

with $\theta(\boldsymbol{t}_a, \boldsymbol{t}_b)$ is the angle between $\boldsymbol{t}_a$ and $\boldsymbol{t}_b$. The probability is estimated by sampling from the set of possible mappings $\boldsymbol{m}$.

5.4 Jaccard Coefficient

Given two subsets $A, B \subset U$ of a universe U the Jaccard coefficient is

$$sim_J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

The hash functions for the LSH scheme are constructed by random orderings of the universe U. Such a random ordering can by viewed as a random permutation π of the elements of U, where $\pi(\cdot)$ delivers the position of an element according to π. The hash function $h_\pi(A) = \min\{\pi(x)\colon x \in A\}$ returns the smallest position of an element of A with respect to the ordering π. Then for two sets A, B :

$$Pr[h_\pi(A) = h_\pi(B)] = sim_J(A,B)$$

The probability is estimated by sampling from the set of possible permutations.

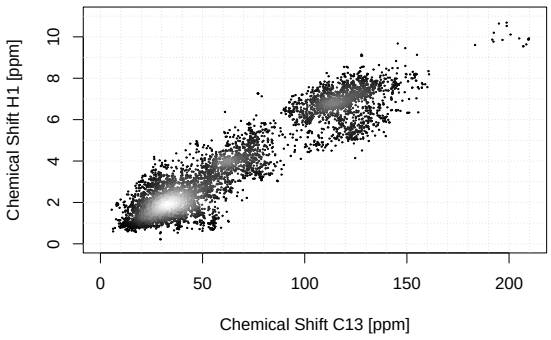


Fig. 5. Density of the peaks of all spectra. Light gray means higher density. Note that when plotting a spectrum with ^{13}C as x-axis (0-220)ppm and ^{1}H as y-axis (0-12)ppm, aromatic structures are located in the upper right region and aliphatic structures are located in lower left region.

6 Results

In this section we evaluate the proposed definition of duplicates and conduct experiments to investigate the tradeoff between costs for candidate filtering of the approximative methods and candidate checking of the exact methods.

6.1 2D-NMR Database

The substances included in the database are mostly secondary metabolites of plants and fungi. They cover a representative area of naturally occurring compounds and originate either from experiments or from simulations[6] based on the known structure of the compound. The database includes 1524 spectra with 2 to 60 peaks each, for a total of about 20,000 peaks. The density in the peak space for all peaks in the database is shown in figure 5.

6.2 Performance Results of the Approximate Methods

We implemented the approximate methods as single SQL statements[7] using the SQL 1999 standard. The used data are the 1524 original spectra, which contain 118 fuzzy duplicates. The run times of the approximate methods are below 20 seconds for all methods. That is a large speedup with respect to the exact methods as well as the heuristics proposed in [15], since those methods run several minutes on that data. The actual speedup depends on the size of the used data set, since the methods of the two classes have different runtime complexities (n^2 versus $n \log n$).

For the approximate methods, we investigate the number of false positives and false negatives for different numbers k of sampled hash functions. First, the parameter $L = 5$ is fixed. For small k more spectra are likely to be reported as similar. The larger k, the more the reported integer vectors as well as the discrete sets have to be identical. Since our mapping to discrete integer vectors

[6] ACD/2D NMR predictor, version 7.08, http://www.acdlabs.com/

[7] The code is available at `http://users.informatik.uni-halle.de/~hinnebur` .

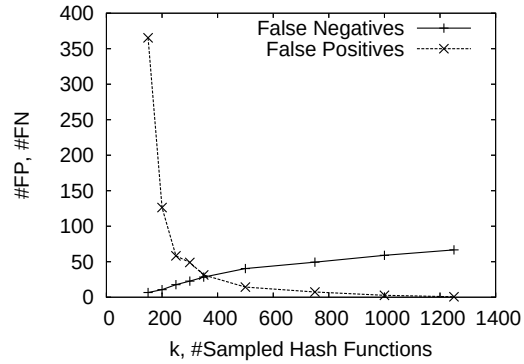


Fig. 6. Number of false positives and false negatives FP,FN for Manhattan with LSH ($L = 5$) and diferent k for four repeated experiments.

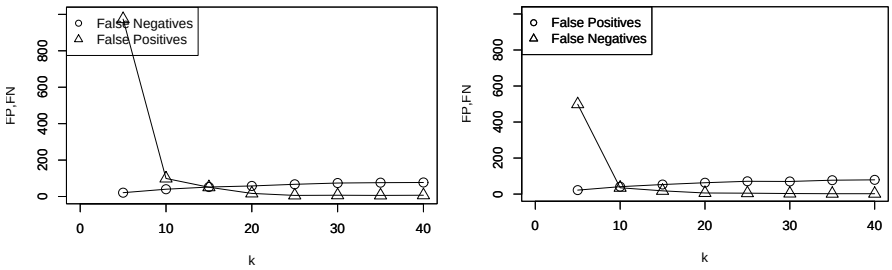


Fig. 7. Number of false positives and false negatives for Jaccard coefficient with Minhashing ($L = 5$), simple grids (left) and shifted grids (right).

and discrete sets respectively may cause false negatives, we want to allow a some variability of the detected spectra.

A relevant performance measure is the number of false positives for very small false negatives. At this point, the reported similar spectra can be subsequently checked with the naive exact method to exclude the false positives. In that respect, the approximate method acts as a strong filter while only few true duplicates are missed. The results for Manhattan distance with LSH are shown in figure 6. Here the number of false positives is about 390 without any false negative. For Jaccard coefficient with Minhashing we tested the mapping to simple grids and shifted grids. The number of false positives are about 900 and 500 respectively, as shown in figure 7.

As Jaccard coefficient with Minhashing gives more false negatives than the Manhattan distance, additionally, we experimented with different values for L. The results are shown in table 1. The table shows (especially in the two blocks at the bottom) that increasing L produces more false positives while the number of false negatives is reduced at the same time.

All reported measurements are averages of five runs. The main point is that merely several hundreds of spectra must be explicitly checked as putative duplicates compared to two millions $(1524 \cdot (1524 - 1)/2)$ for the naive method. For

Table 1. Number of false positives and false negatives for Jaccard coefficient with Minhashing for different setting for L and k.

k	L	Minhashing		Minhashing+Shift	
		FN	FP	FN	FP
2	1	42	9352	46	2918
3	1	59	252	55	558
4	1	67	170	57	168
5	1	69	57	66	47
2	5	19	15167	11	13828
3	5	32	2626	31	1540
4	5	39	514	36	547
5	5	46	199	47	183
5	10	35	444	31	285
5	15	26	654	17	481
5	20	25	836	16	584
5	50	20	1445	12	1119

comparison, the best exact heuristic reported in [15] still needs to check about 30,000 duplicate pairs with the naive method. So, approximate methods are a huge performance gain.

In conclusion, the mapping to integer vector in combination with Manhattan distance and LSH turned out to be the best method, delivering the least number of false positives and no false negatives. The mapping to shifted grids is better than the mapping to simple grids, but the number of false positives is higher. However, the minhashing method has a slight runtime advantage, since less hash functions need to be sampled. This might be useful in case of very large data sets.

6.3 Detected Duplicates

There were no duplicates intentionally included in the database. With a setting of $\alpha = 3$ppm and $\beta = 0.3$ppm, which are reasonable tolerances, 118 of 2,322,576 (naive method) possible pairs are reported as fuzzy duplicates.

The found duplicate pairs revealed the following types of classes of duplicates occurring in practice: (i) accidental entry of the same spectra/substance with different names, (ii) spectra prediction software ignoring stereochemical quaternary carbon configurations, (iii) some pairs consist of an experimental and a simulated spectrum (see figure 8) of the same substance (which speaks for both our duplicate definition and the simulation software), (iv) same chemical compound in different measurement conditions (measurement frequency, solvent).

Due to the deletion of peaks in the preprocessing step, different substitutional patterns are also candidates for near duplicates because a discrimination between a peak splitting event or an additional substituent peak is not possible.

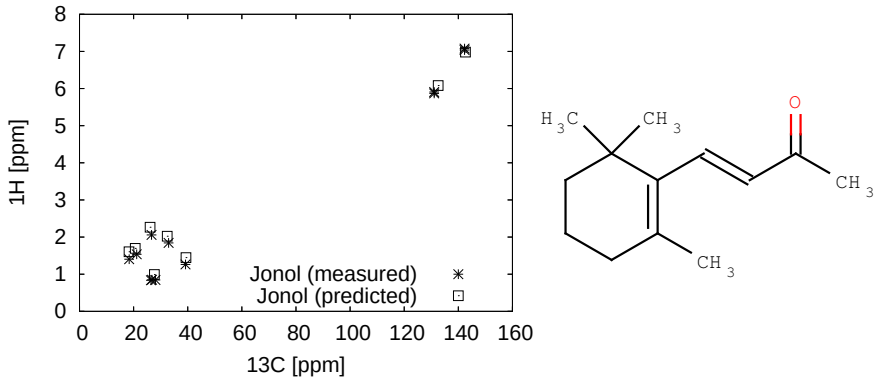


Fig. 8. Two spectra as an example for a detected duplicate in our database: Peaks as simple points from an experimental and predicted spectrum of β –Jonol. Note, that each peak in A has matching peak in B according to $\alpha = 3.0ppm$ and $\beta = 0.3ppm$.

7 Conclusion

We proposed a simple and robust definition for fuzzy duplicates of 2D-NMR spectra on the basis of co-matching peaks. Considering peak splitting as well as inherent measurement errors are crucial to respect for in NMR- Data. We described ideas and heuristics to embed 2D- spectra data into vector spaces and discrete objects, to suitably interface NMR- data to data mining algorithms. A scale up to large data volumes is achieved by applying approximate and fast algorithms as preliminary filters prior to the computation of the exact duplicates, avoiding the quadratic nature of searching for duplicates in sets of spectra.

We found that our mapping to integer vectors in combination with LSH and Manhattan distance is more suitable for the task than mappings to discrete set in combination with Jaccard coefficient and minhashing. A conservative choice of the parameters guarantees no false negatives. The developed methods are the foundation to start and manage a large collection of NMR spectra, which is part of an ongoing metabolomics project at the IPB in Halle (Saale).

Acknowledgements

Thanks to Andrea Porzel for valuable discussions and access to the NMR data collection. Steffen Neumann is supported under BMBF grant 0312706G.

References

1. A. Tsipouras, J. Ondeyka, C. Dufresne et al. Using similarity searches over databases of estimated c-13 nmr spectra for structure identification of natural products. *Analytica Chimica Acta*, 316:161–171, 1995.

2. A. S. Barros and D. N. Rutledge. Segmented principal component transform-principal component analysis. *Chemometrics & Intelligent Laboratory Systems*, 78:125–137, 2005.
3. M. Bilenko and R. J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 39–48, New York, NY, USA, 2003. ACM Press.
4. A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the web. In *Selected papers from the sixth international conference on World Wide Web*, pages 1157–1166, Essex, UK, 1997. Elsevier Science Publishers Ltd.
5. A. Chowdhury, O. Frieder, D. Grossman, and M. C. McCabe. Collection statistics for fast duplicate document detection. *ACM Trans. Inf. Syst.*, 20(2):171–191, 2002.
6. E. Cohen. Size-estimation framework with applications to transitive closure and reachability. *J. Comput. Syst. Sci.*, 55(3):441–453, 1997.
7. J. D. Cohen, M. C. Lin, D. Manocha, and M. K. Ponamgi. I-COLLIDE: An interactive and exact collision detection system for large-scale environments. In *Symposium on Interactive 3D Graphics*, pages 189–196, 218, 1995.
8. J. G. Conrad, X. S. Guo, and C. P. Schriber. Online duplicate document detection: signature reliability in a dynamic retrieval environment. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 443–452, New York, NY, USA, 2003. ACM Press.
9. F. Deng and D. Rafiei. Approximately detecting duplicates for streaming data using stable bloom filters. In *SIGMOD '06: Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 25–36, New York, NY, USA, 2006. ACM Press.
10. A. Gionis, D. Gunopulos, and N. Koudas. Efficient and tunable similar set retrieval. In *SIGMOD '01: Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, pages 247–258, New York, NY, USA, 2001. ACM Press.
11. A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *VLDB'99: Proceedings of the 25th International Conference on Very Large Data Bases*, pages 518–529, CA USA, 1999. Morgan Kaufmann Publishers Inc.
12. D. Gomes, A. L. Santos, and M. J. Silva. Managing duplicates in a web archive. In *SAC '06: Proceedings of the 2006 ACM symposium on Applied computing*, pages 818–825, New York, NY, USA, 2006. ACM Press.
13. M. Henzinger. Finding near-duplicate web pages: a large-scale evaluation of algorithms. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 284–291, New York, NY, USA, 2006. ACM Press.
14. M. A. Hernandez and S. J. Stolfo. Real-world data is dirty: Data cleansing and the merge/purge problem. *Data Mining and Knowledge Discovery*, 2(1):9–37, 1998.
15. A. Hinneburg, B. Egert, and A. Porzel. Duplicate detection of 2d-nmr spectra. *Journal of Integrative Bioinformatics*, 4(1):53, 2007.
16. P. Indyk and R. Motwani. Approximate nearest neighbor - towards removing the curse of dimensionality. In *Proceedings of the 30th Symposium on Theory of Computing*, pages 604–613, 1998.
17. Y. Ke, R. Sukthankar, and L. Huston. An efficient parts-based near-duplicate and sub-image retrieval system. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 869–876, New York, NY, USA, 2004. ACM Press.

18. P. Krishnan, N. J. Kruger, and R. G. Ratcliffe. Metabolite fingerprinting and profiling in plants using nmr. *Journal of Experimental Botany*, 56:255–265, 2005.
19. M. Farkas, J. Bendl, D. H. Welti et al. Similarity search for a h-1 nmr spectroscopic data base. *Analytica Chimica Acta*, 206:173–187, 1988.
20. A. Metwally, D. Agrawal, and A. E. Abbadi. Duplicate detection in click streams. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 12–21, New York, NY, USA, 2005. ACM Press.
21. G. N. Noren, R. Orre, and A. Bate. A hit-miss model for duplicate detection in the who drug safety database. In *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 459–468, New York, NY, USA, 2005. ACM Press.
22. C. Steinbeck, S. Krause, and S. Kuhn. Nmrshiftdb-constructing a free chemical information system with open-source components. *J. chem. inf. & comp. sci.*, 43:1733 –1739, 2003.
23. M. Weis and F. Naumann. Detecting duplicate objects in xml documents. In *IQIS '04: Proceedings of the 2004 international workshop on Information quality in information systems*, pages 10–19, New York, NY, USA, 2004. ACM Press.
24. H. Yang and J. Callan. Near-duplicate detection by instance-level constrained clustering. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 421–428, New York, NY, USA, 2006. ACM Press.