Using Consecutive Support for Genomic Profiling*

Edgar H. de Graaf, Jeannette de Graaf, and Walter A. Kosters

Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands
{edegraaf,graaf,kosters}@liacs.nl

Abstract. We propose a new measure of support (the number of occurrences of a pattern), in which instances are more important if they occur with a certain frequency and close after each other in the stream of records. We will explain this new consecutive support and show how consecutiveness and the notion of hypercliques can be incorporated into the ECLAT algorithm.
Synthetic examples show how interesting phenomena can now be discovered in the datasets. The new measure can be applied in many areas, ranging from bio-informatics to trade, supermarkets, and even law enforcement. We will use it in genomic profiling, where it is important to find patterns contained in many individuals: patterns close together in one chromosome are more significant.

1 Introduction

In earlier research we explored the use of frequent itemsets to visualize deviations in chromosome data concerning people with a certain illness, genomic profiling [4]. During our exploration of this problem it became apparent that patterns are more important when the areas (transactions) in which they occur are close together. The consecutiveness of transactions containing the pattern plays an important role in other applications too. Patterns are frequent sets of items, where frequent means that their support, that can be defined in different ways, is more than a pre-given threshold. In the biological problem the items are individuals and the transactions are "clones", pieces of the chromosome that might occur more or less often than in a healthy individual. Patterns in close transactions are better because they are close together in the chromosome and are biologically more significant than patterns that are far apart and in different chromosomes.

Consecutive support informally is the support or the number of occurrences of patterns where we take into account the distance between transactions that contain the pattern: the consecutive support should be higher when occurrences are close together. Here distance is the number of in-between transactions that do not contain the pattern. Of course, this only makes sense if the transactions

* This research is carried out within the Netherlands Organization for Scientific Research (NWO) MISTA Project (grant no. 612.066.304).

are given in some logical order. We will use consecutive support for genomic profiling, however this type of support can be applied in a number of other domains:

- **Supermarket**. E.g., big supermarkets receive large quantities of goods every day. Knowing which goods will be sold in large quantities close in time helps the supermarket decide when to refill these goods.
- **Trading**. E.g., a combination of stock being sold once may lead to waves of these stocks being sold close after each other while other combinations might not.
- **Law enforcement**. E.g., when police investigates telephone calls, subjects that are discussed during a longer period might be more interesting than subjects (word combinations) that are mentioned often at separate moments.

In this paper we define consecutive support, having two parameters: the reward factor ρ and the punishment factor σ. Existing pruning methods can be easily incorporated. In particular h-confidence and hypercliques enable us to amplify consecutive behaviour. With our experimental results we show how consecutive support, compared to the results in [4], gives new and interesting patterns when applied to the biological problem of finding patterns in chromosomes.

This research is related to work done on the (re)definition of support and gap constraint. The notion of support was first introduced by Agrawal et al. in [1] in 1993. Much later Steinbach et al. in [8] generalized the notion of support providing a framework for different definitions of support in the future. Our notion of consecutive support is not easily fitted in the eval-function provided there. (Next to this framework Steinbach also provides a couple of example functions.) Frequent itemset mining on similar data was done by Rouveirol et al. in [7]. Our work is related to this work because of the minimal frequency constraint also used in consecutive support.

If we take the database of clones as an example, we have a database where the clones (or transactions) are itemsets of patients with gains or losses in the clones. We could transpose this database so that transactions correspond to the patients, and are itemsets of clones that showed gains or losses. Now we can search for patterns and with techniques like the time window constraint as defined in [5] or the gap constraint as defined in [2], we can search for clones that are close together in the chromosomes. However, the combination of patients with equal clones will be lost.

Finally this work is related to some of our earlier work. Primarily the work done in [4] already stated that the biological problem could profit from incorporating consecutiveness into frequent itemset mining. Secondly in [3] it was mentioned that support is just another measure of saying how good a pattern fits with the data. There we defined different variations of this measure, and consecutive support can been seen as such a variation.

The formal definitions concerning consecutive support are given in Section 2. A particular pruning method is discussed in Section 3. In Section 4 we present experimental results, and we conclude in Section 5.

2 Consecutive Support

2.1 Definition

The definition of association rules relies on that of support: the number of transactions that contain a given itemset. In this paper we propose a more general definition, that takes the consecutiveness of the transactions into account.

Suppose items are from the set $\mathcal{I} = \{1, 2, \ldots, n\}$, where $n \geq 1$ is a fixed integer constant. A *transaction* is an *itemset*, which is a subset of $\mathcal{I}$. A *database* is an *ordered series* of m transactions, where $m \geq 1$ is a fixed integer constant. If an itemset is an element of a database, it is usually referred to as a transaction.

The *traditional support* of an itemset I with respect to a database $\mathcal{D}$, denoted by $\text{TradSupp}(I, \mathcal{D})$, is the number of transactions from $\mathcal{D}$ that contain I. Clearly, $0 \leq \text{TradSupp}(I, \mathcal{D}) \leq m$.

An important property of the traditional support is the so-called APRIORI property [1] or anti-monotonicity constraint: if itemset I is contained in itemset I', the support of I is larger than or equal to the support of I'. We want the new measure to satisfy this constraint also.

The support measure we propose is a generalization of the traditional support. In order to take into account the consecutiveness of a pattern we use two real parameters $\rho \geq 0$ and $0 \leq \sigma \leq 1$. With ρ we reward the pattern if it occurs in consecutive transactions, with σ we punish for the gaps between the consecutive occurrences of the pattern in the database.

Suppose we have an itemset I and let $O_j \in \{0, 1\}$ $(j = 1, 2, \ldots, m)$ denote whether or not the j^{th} transaction in the database $\mathcal{D}$ contains I (O_j is 1 if it does contain I, and 0 otherwise; the O_j's are referred to as the *O-series*). The following algorithm computes a real value t in one linear sweep through the database and the resulting t is defined as the *consecutive support* of I with respect to $\mathcal{D}$ (denoted by $\text{Supp}(I, \mathcal{D}, \rho, \sigma)$):

$t := 0$; $j := 1$; $\mathit{reward} := 0$;
while ($j \leq m$) **do**
  if ($O_j = 1$) **then**
    $t := t + 1 + \mathit{reward}$; $\mathit{reward} := \mathit{reward} + \rho$;
  else
    $\mathit{reward} := \mathit{reward} \cdot \sigma$;
  fi
  $j := j + 1$;
od

The consecutive support t can become very large, and one could for example use $\sqrt{t}$ instead. In our examples we will always employ just t.

Example 1. Assume the O-series of a certain pattern I equals 101101, $\rho = 1$ and $\sigma = 0.1$. The consecutive support t will then be 5.41:

O	1	0	1	1	0	1
reward	0	1	0.1	1.1	2.1	0.21
t	1	1	2.1	4.2	4.2	**5.41**

2.2 Formal Discussion

During the loop the value of *reward*, which "rewards" the occurrence of a 1, is always at least 0. If *reward* would never be adapted, i.e., it would remain 0 all the time, independent of the itemset I, the algorithm would compute $\mathrm{TradSupp}(I, \mathcal{D})$. This is the case when $\rho = 0$: $\mathrm{Supp}(I, \mathcal{D}, 0, \sigma) = \mathrm{TradSupp}(I, \mathcal{D})$ for any $0 \leq \sigma \leq 1$. So the consecutive support is indeed a generalization of the traditional support. Furthermore we have: for all $\rho \geq 0$ and $0 \leq \sigma \leq 1$, $\mathrm{Supp}(I, \mathcal{D}, \rho, \sigma) \geq \mathrm{TradSupp}(I, \mathcal{D})$.

It is clear that the APRIORI property is satisfied: for all $\rho \geq 0$ and $0 \leq \sigma \leq 1$, $\mathrm{Supp}(I, \mathcal{D}, \rho, \sigma) \geq \mathrm{Supp}(I', \mathcal{D}, \rho, \sigma)$ if the itemset I' contains the itemset I. This follows from the observation that the *reward*-values in the I'-case are never larger than those in the I-case.

Finally, we easily see that $0 \leq \mathrm{Supp}(I, \mathcal{D}, \rho, \sigma) \leq m + m(m-1)\rho/2$. The maximum value is obtained if and only if all transactions from the database $\mathcal{D}$ contain I, i.e., an O-series entirely consisting of 1s. Only the all 0s series gives the minimum value 0.

It is not hard to show that for the O-series $1^{a_1}0^{b_1}1^{a_2}0^{b_2}\ldots 0^{b_{n-1}}1^{a_n}$ (a series of a_1 1s, b_1 0s, a_2 1s, b_2 0s, $\ldots$, b_{n-1} 0s, a_n 1s) the consecutive support equals

$$\sum_{i=1}^{n} a_i + \rho \sum_{i=1}^{n} a_i(a_i-1)/2 + \rho \sum_{1\leq i<j\leq n} a_i a_j \sigma^{b_i+b_{i+1}+\cdots+b_{j-1}} =$$
$$(1-\rho/2)S + \rho S^2/2 - \rho \sum_{1\leq i<j\leq n} a_i a_j (1-\sigma^{b_i+b_{i+1}+\cdots+b_{j-1}}),$$

where $S = \sum_{i=1}^{n} a_i$ (i.e., the traditional support); here 0^0 must be interpreted as 1 (an exponent 0 can be avoided by demanding all b_i's to be non-zero; if we also demand all a_i's to be > 0, both the number n and the numbers a_i and b_i are unique, given an O-series). The formula follows from the fact that if *reward* equals ε, then the series 1^k0^ℓ changes this into $(\varepsilon + k\rho)\cdot\sigma^\ell$, meanwhile giving a contribution of $k + k\varepsilon + k(k-1)\rho/2$ to the consecutive support. An extra series 0^ℓ at the beginning or end has no influence on the consecutive support.

The second part of the equation, $\rho \sum_{i=1}^{n} a_i(a_i-1)/2$, consists of the ρs added for a subset of consecutive 1s in the O-series. The last part of the equation is the addition of the rewards from the previous consecutive 1s decreased by σ, because of the number of 0s between the groups of consecutive 1s. Also note that when we choose $\rho = 2$ we get $S^2 - \rho \sum_{1\leq i<j\leq n} a_i a_j (1-\sigma^{b_i+b_{i+1}+\cdots+b_{j-1}})$. This shows that consecutive support is at most S^2 if $\rho = 2$.

Example 2. Take $\rho = 2$. Then the O-series $1^5 0^\ell 1^4$ has consecutive support $81 - 40(1-\sigma^\ell)$. Note that this is the same for the reverse $1^4 0^\ell 1^5$. As $\ell \to \infty$ this value approaches $41 = 5^2 + 4^2$, whereas for small ℓ and $\sigma \approx 1$ it is near $81 = (5+4)^2$.

It can be observed that the consecutive support as defined above only depends on the lengths of the "runs" and the lengths of the intermediate "non-runs": the a_i's and b_i's above. Here a *run* is defined as a maximal consecutive series of 1s in

a 0/1 sequence. Indeed, the sum $\sum_{k=i}^{j-1} b_k$ equals the number of 0s between run i and run j. This also implies that the definition is *symmetric*, in the sense that the support is unchanged if the order of the O-series is reversed — a property that is certainly required.

The reason why we add ρ and multiply by σ instead of, for example, add ρ and subtract σ, lies in the observation that in the latter case the symmetry property would not hold. Subtracting σ leads to different consecutive support values for an O-series and its reverse. E.g., if $\rho = 2$ and $\sigma = 0.5$, $1^5 0^3$ would give 25, whereas $0^3 1^5$ has 17.5 (the definition from Section 2.1 gives 25 in both cases). One should also take care that the support remains positive in that case.

Instead of this way of calculating consecutive support it is also possible to augment the O-series with *time stamps*. Then one is able to use the real time between two transactions in calculating the consecutive support. In the previous definition each transaction was assumed to take the same amount of time. Another improvement might be to reinitialize *reward* to 0 at suitable moments, for instance at chromosome boundaries or at "closing hours".

We consider algorithms that find all frequent itemsets, given a database. A *frequent* itemset is an itemset with support at least equal to some pre-given threshold, the so-called *minsup*. Thanks to the APRIORI property many efficient algorithms exist. However, the really fast ones rely upon the concept of FP-TREE or something similar, which does not keep track of consecutivity. This makes these algorithms hard to adapt for consecutive support.

One fast algorithm that does not make use of FP-TREEs is called ECLAT [10]. ECLAT grows patterns recursively while remembering which transactions contained the pattern, making it very suitable for consecutive support. In the next recursive step only these transactions are considered when counting the occurrence of a pattern. All counting is done by using a matrix and patterns are extended with new items using the order in the matrix. It is straightforward to adapt ECLAT to incorporate consecutiveness, the counting of traditional support is simply replaced by the $\mathrm{Supp}(I, \mathcal{D}, \rho, \sigma)$ function as proposed earlier. The overhead of extra calculations is minimal and the runtime complexity is expected to be equal to that of ECLAT as described in [10].

3 Hyperclique Patterns and h-confidence

Many pruning principles used for traditional support calculation can still be applied for consecutive support. We consider one method in particular. In the case of our major example, the database of clones, we wanted to visualize patterns with a certain minimal consecutive support. Unfortunately there are many patterns with this support. In order to speed up the search and to filter out uninteresting patterns we can search for *hyperclique patterns* as described in [9]. Because of space limitations we explain hyperclique patterns via an example:

Example 3. First a *minimal confidence threshold* h_c is defined, say $h_c = 0.6$. We want to know if $\{A, B, C\}$ is a hyperclique pattern. We calculate the confidence of $A \to \{B, C\}$, $B \to \{A, C\}$ and $C \to \{A, B\}$. The lowest of these

confidences is the *h-confidence*, which must be higher then h_c. Assume that $conf(A \rightarrow B, C) = \text{Supp}(\{A, B, C\}, \mathcal{D}, \rho, \sigma)$ / $\text{Supp}(\{A\}, \mathcal{D}, \rho, \sigma) = 0.58$. Then $\{A, B, C\}$ is no hyperclique pattern.

When we combine the concept of consecutive support with hyperclique patterns we get patterns that occur frequent but in the flow of transactions close after each other and there is a *strong affinity* between items: the presence of $x \in P$, where P is an itemset or clone, in a transaction strongly implies the presence of the other items or patients in P.

It is clear that hyperclique patterns possess the *cross-support property*. This means that we will not get *cross-support patterns*. These are patterns containing items of substantially different support levels. If one item has a high support and another item has a low support, then h-confidence will be low if the denominator is the item with the high support.

Example 4. Say A is an item with a consecutive support of 200 and B has a consecutive support of 50. The support of $\{A, B, C\}$ will at most be 50 because of the APRIORI property. So $conf(A \rightarrow B, C)$ can at most be $50/200 = 0.25$. As a consequence the h-confidence of $\{A, B, C\}$ will also be at most 0.25. So if $h_c = 0.6$, then $\{A, B, C\}$ and all the patterns that are grown from it can be pruned.

The combination of hyperclique patterns and consecutive support allows us to find patterns that occur in clones (transactions) that follow each other close, yet minimal support can be relatively low. This property is especially handy for our motivating example, because a low minimal consecutive support will generate many cross-support patterns, which are pruned if we search only for hyperclique patterns. Hyperclique patterns also posses the anti-monotonicity property, because as patterns grow the numerator of the confidence calculation stays the same or declines. The denominator stays fixed and so h-confidence will decrease or stay the same:

Example 5. Say $conf(A \rightarrow B, C) = 0.58$. The superset $\{A, B, C, D\}$ will at most have the same consecutive support as $\{A, B, C\}$. Also the denominator $\text{Supp}(\{A\}, \mathcal{D}, \rho, \sigma)$ stays the same, so the h-confidence of $\{A, B, C, D\}$ can at most be 0.58.

4 Results and Performance

The experiments were done for three main reasons. First of all we want to show that consecutive support can enable one to find new patterns that one does not find with the traditional support. Secondly we want to show how using the principle of h-confidence one can filter the data. Finally we want to give an indication how the reward factor ρ and punishment factor σ should be chosen.

All experiments were done on a Pentium 4 2.8 GHz with 512MB RAM. For our experiments we used five datasets. One biological dataset, referred to as the

Nakao dataset, was also used in [4]. This data set originates from Nakao et al. who used the dataset in [6]. This publicly available dataset contains normalized $\log_2$-ratios for 2124 clones, located on chromosomes 1–22 and the X-chromosome. Each clone is a transaction with 2 to 1020 real numbers corresponding to patients. We can look at gains and/or losses. If we consider gains, a patient is present in a transaction (clone) if his value is at least 0.225 higher than that of a healthy person (for losses at least 0.225 lower). The work in this paper reported losses and gains in chromosomes 1, 8, 17, 18 and 20. Two datasets are synthetic databases, but structured like the dataset of clones. One of these datasets, the *noisy dataset*, contains more noise then the other, the *ideal dataset*. The precise structure of these datasets is described in [4]. The remaining datasets are synthetic datasets made to show how consecutive support can be used to find patterns that could not be found before. The third synthetic data set, referred to as the *food+drink dataset*, describes a cafe-restaurant where in the middle of a day a lot of people buy bread and orange juice; it has 1000 transactions (customers) and 100 items (products). The fourth synthetic data set will be explained later.

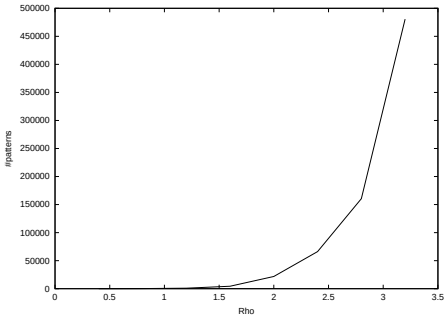

Fig. 1. Number of patterns from the Nakao dataset as ρ increases (gains, $minsup = 625$, $\sigma = 0.5$)

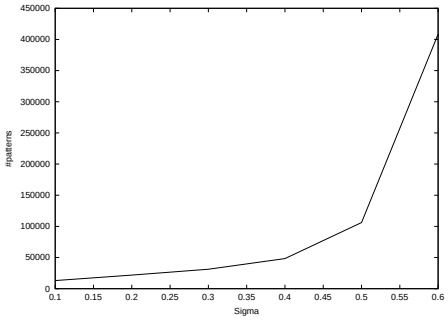

Fig. 2. Number of patterns from the Nakao dataset as σ increases (gains, $minsup = 625$, $\rho = 2.0$)

4.1 Consecutive Support

Figure 1 and Figure 2 show how the number of patterns increases with ρ and σ. Each setting therefore requires another *minsup*. In some cases it is best to select the *minsup* such that one gets a fixed number of patterns, e.g., 1000, in order to compare the results.

In the experiments of Figure 3–6 we tried to find approximately 1000 patterns with the highest traditional or consecutive support. After this we count for each transaction how many patterns it contains, allowing us to see how active areas are. For the Nakao dataset more active means that many clones (gains) in the same area are present in many groups of patients.

Figure 3 and Figure 5 show where patterns occur when we use traditional support, giving results similar to those in [4]. For each transaction the number

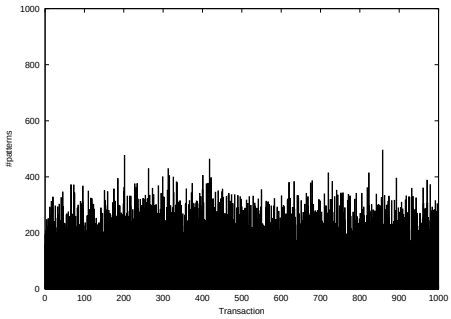

Fig. 3. Occurrence graph of food+drink using traditional support ($minsup = 257$)

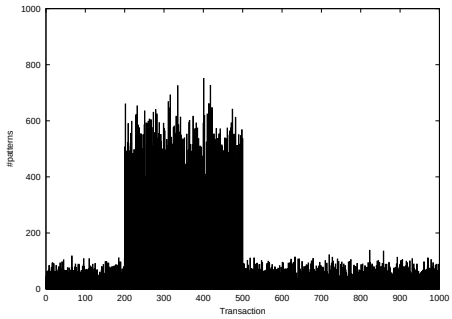

Fig. 4. Occurrence graph of food+drink using consecutive support ($minsup = 467$, $\rho = 1.0$ and $\sigma = 0.5$)

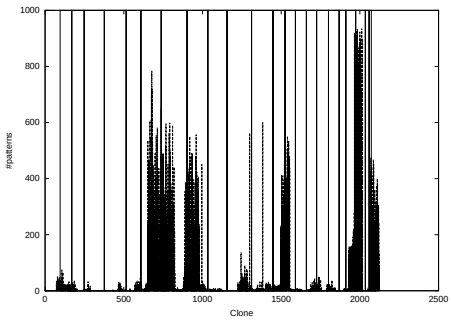

Fig. 5. Occurrence graph of Nakao using traditional support (gains, $minsup = 129$)

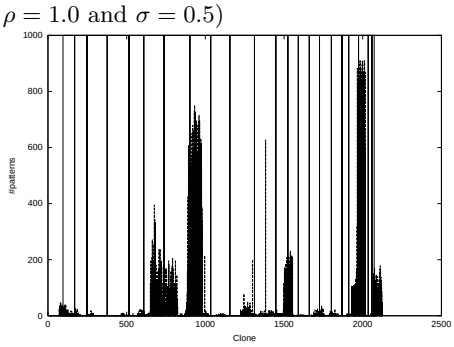

Fig. 6. Occurrence graph of Nakao using consecutive support (gains, $minsup = 827$, $\rho = 1.0$ and $\sigma = 0.5$)

of patterns that it occurs in is plotted in a so-called *occurrence graph*. In each of these graphs we will indicate chromosome borders when the Nakao dataset is visualized. In the food+drink dataset it is very clear that consecutive support enables us to see new patterns. Figure 4 shows that in certain areas patterns are more consecutive. Figure 6 shows that certain areas are less active if we use consecutive support instead of traditional support (chromosomes 7 and 8) and some areas contain more patterns (chromosome 9), hence providing patterns that occur together in one part of the chromosome instead of far apart. This shows additional activity compared to results reported by Nakao et al. in [6].

In order to evaluate the effect of more or less noise on consecutive support we used the ideal and noisy dataset. These datasets are generated with properties similar to the Nakao dataset with real patient information (see [4] for details). The results for the ideal dataset are plotted in Figure 7 and 8.

Figure 7 shows that some interesting areas are less clear when using traditional support. However they become more apparent when we apply consecutive support. The results for the noisy dataset are displayed in Figure 9 and 10. Because of the noise the middle peak becomes less clear. However overall the results seem hardly to be affected by noise.

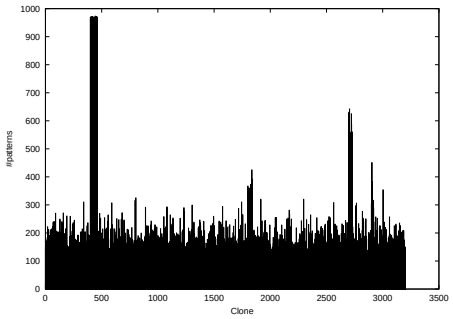

Fig. 7. Occurrence graph of the ideal dataset using traditional support (gains, $minsup = 479$)

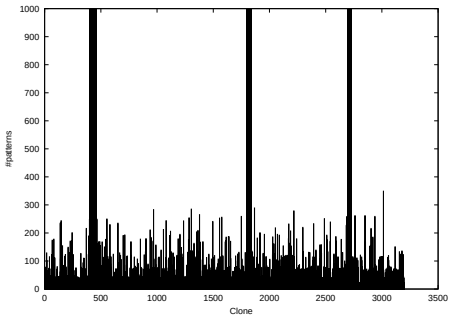

Fig. 8. Occurrence graph of the ideal dataset using consecutive support (gains, $minsup = 6180$, $\rho = 2.0$ and $\sigma = 0.7$)

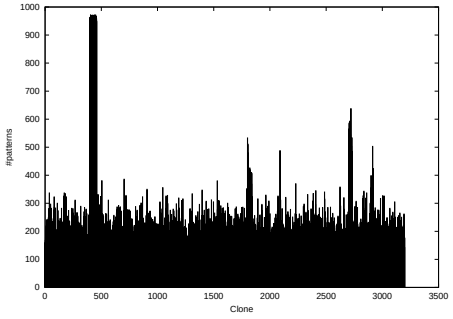

Fig. 9. Occurrence graph of the noisy dataset using traditional support (gains, $minsup = 617$)

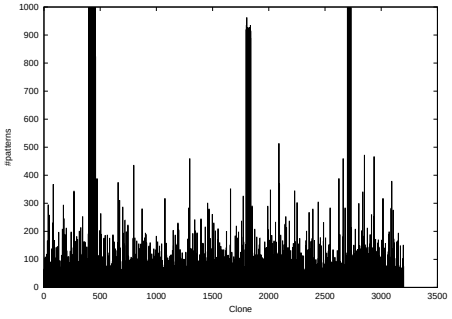

Fig. 10. Occurrence graph of the noisy dataset using consecutive support (gains, $minsup = 6039$, $\rho = 2.0$, $\sigma = 0.7$)

4.2 Selection of ρ and σ

The goal of the following experiments was to give some guidance in the selection of reward factor ρ and punishment factor σ. The right parameters should result in many patterns of which the O-series has large groups of consecutive 1s.

Figure 11 plots the average number of consecutive groups of 1s and 0s for all patterns. The plot gives an indication of consecutiveness of patterns found using different settings of ρ and σ (less groups indicate more consecutiveness). The plot seems to stabilize around $\rho = 2$. Figure 12 and 13 show that only if we choose σ very close to 1.0 we get results more like those for traditional support. However, Figure 13 still shows some influence of ρ. For the Nakao dataset it seems that if $\rho \approx 2$, then the influence of σ is minimalized as long as σ is not too close to 1.0. Also similar experiments showed significant changes in the occurrence graph only if ρ was chosen very small. Different datasets might require different settings depending on how much one wants to amplify consecutiveness. However results in this section indicate that $\rho = 2.0$ and $0.2 \leq \sigma \leq 0.8$ seem to be good choices. However, a lot of experimental work is necessary to settle this issue.

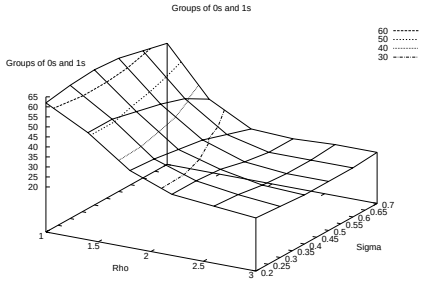

Fig. 11. Effect of ρ and σ on the O-series for the Nakao dataset (gains, $minsup = 625 \cdot (\rho/2)$, chosen to guarantee a reasonable amount of patterns)

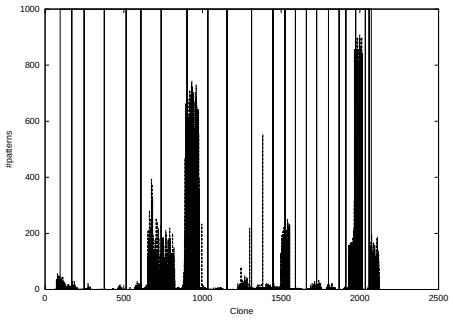

Fig. 12. Occurrence graph of Nakao using consecutive support (gains, $minsup = 2498$, $\rho = 2.0$ and $\sigma = 0.8$)

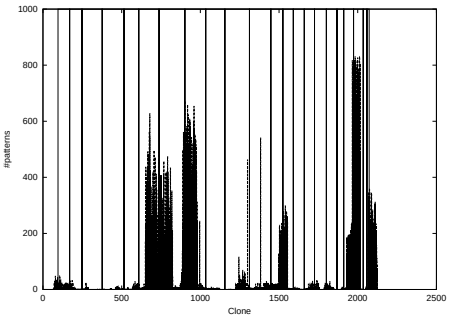

Fig. 13. Occurrence graph of Nakao using consecutive support (gains, $minsup = 6157$, $\rho = 2.0$ and $\sigma = 0.99$)

4.3 Combination with h-confidence

In the following experiments the goal was to show that combining hyperclique patterns with consecutive support enables us to see patterns occurring in bursts. In order to show this we created a new synthetic dataset, referred to as the *coffee+cookie dataset*, where in the cafe-restaurant small bursts of people buy coffee and a cookie, during the day in the coffee breaks.

Figure 14 does not show the small groups buying the same products: just hyperclique patterns do not reveal the bursts. Figure 15 shows that with only consecutive support we are also unable to discover these patterns. Figure 16 shows people buying the products in bursts. Consecutive support stresses patterns that are consecutive and the principle of h-confidence filters out the noise caused by cross-support patterns.

When we apply these techniques to the Nakao dataset (losses), in Figure 18, we can see, e.g., on chromosomes 14 and 15 (near transaction 1600) that certain areas become more active compared to not using h-confidence in Figure 17.

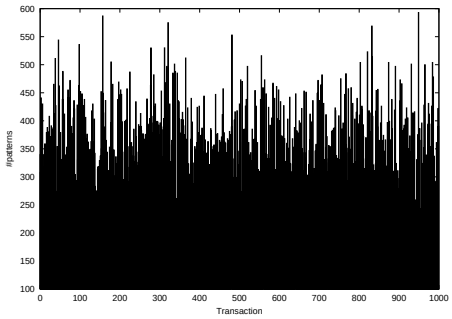

Fig. 14. Occurrence graph of coffee+cookie using only h-confidence ($minsup = 64$, $h_c = 0.5$)

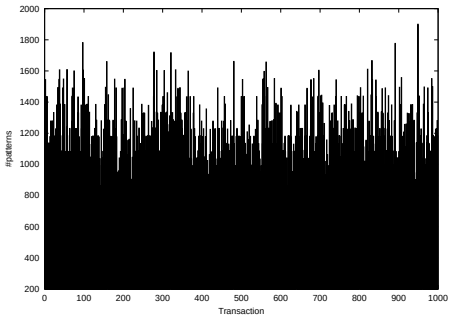

Fig. 15. Occurrence graph of coffee+cookie using only consecutive support ($minsup = 225$, $\rho = 1.0$, $\sigma = 0.5$, $h_c = 0$)

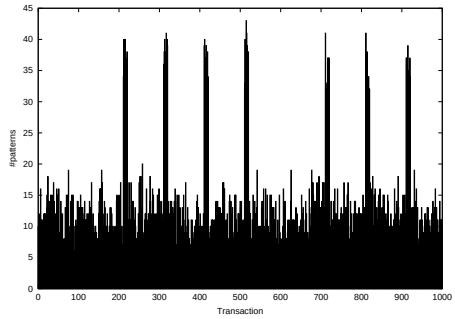

Fig. 16. Occurrence graph of coffee+cookie using both consecutive support and h-confidence ($minsup = 64$, $\rho = 1.0$, $\sigma = 0.5$, $h_c = 0.31$)

5 Conclusions and Future Work

Consecutive support enables us to find new and useful patterns in the chromosomes compared to methods using only traditional support. Principles applicable to traditional support can still be used with consecutive support. For instance the combination of consecutive support and the h-confidence threshold enables us to find small bursts of patterns. In this case h-confidence filters out noise and consecutive support amplifies the bursts.

Using the distance between transactions like it is done in this paper is an interesting area of research. In the future we want to examine if consecutive support enables us to visualize even more types of pattern occurrence, perhaps even detecting them automatically. Also we want to see if we can speed up the search for consecutive patterns. Finally we want to extend consecutive support by using distance between transactions in different ways, which might give us even more biological relevant patterns.

Acknowledgments We would like to thank Joost Broekens, Joost Kok, Siegfried Nijssen and Wim Pijls.

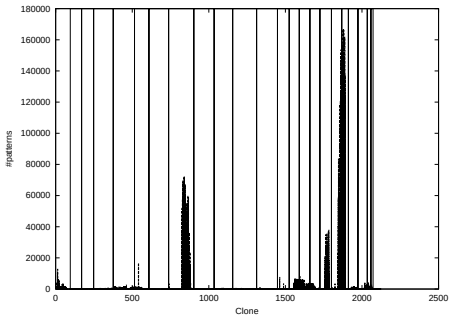

Fig. 17. Occurrence graph of Nakao: consecutive support (losses, $minsup = 400$, $\rho = 1.0$, $\sigma = 0.9$, $h_c = 0$)

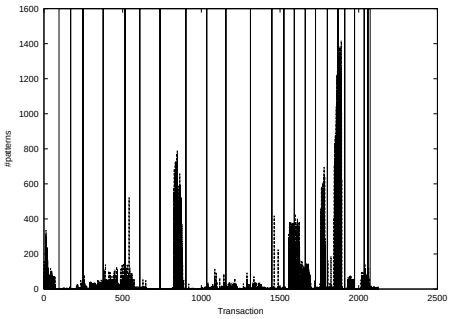

Fig. 18. Occurrence graph of Nakao: consecutive support and h-confidence (losses, $minsup = 25$, $\rho = 1.0$, $\sigma = 0.9$, $h_c = 0.15$)

References

1. Agrawal, R., Imielinski, T., Srikant, R.: *Mining Association Rules between Sets of Items in Large Databases.* In Proc. of ACM SIGMOD Conference on Management of Data (1993), pp. 207–216.
2. Antunes, C., Oliveira, A.L.: *Generalization of Pattern-Growth Methods for Sequential Pattern Mining with Gap Constraints.* In Machine Learning and Data Mining in Pattern Recognition (MLDM 2003), LNCS 2734, Springer, pp. 239–251.
3. Graaf, E.H. de, Kosters, W.A.: *Using a Probable Time Window for Efficient Pattern Mining in a Receptor Database.* In Proc. of 3rd Int. ECML/PKDD Workshop on Mining Graphs, Trees and Sequences (MGTS'05), pp. 13–24.
4. Graaf, J.M. de, Menezes, R.X. de, Boer, J.M., Kosters, W.A.: *Frequent Itemsets for Genomic Profiling.* In Proc. 1st International Symposium on Computational Life Sciences (CompLife 2005), LNCS 3695, Springer, pp. 104–116.
5. Leleu, M., Rigotti, C., Boulicaut, J.F., Euvrard, G.: *Constraint-Based Mining of Sequential Patterns over Datasets with Consecutive Repetitions.* In Proc. 7th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2003), LNAI 2838, Springer, pp. 303–314.
6. Nakao, K., Mehta, K.R., Fridlyand, J., Moore, D.H., Jain, A.N., Lafuente, A., Wiencke, J.W., Terdiman, J.P., Waldman, F.M.: *High-Resolution Analysis of DNA Copy Number Alterations in Colorectal Cancer by Array-Based Comparative Genomic Hybridization.* Carcinogenesis 25 (2004), pp. 1345–1357.
7. Rouveirol, C., Stransky, N., Hupé, Ph., La Rosa, Ph., Viara, E., Barillot, E., Radvanyi, F.: *Computation of Recurrent Minimal Genomic Alterations from Array-CGH Data.* Bioinformatics 22 (2006), pp. 849–856.
8. Steinbach, M., Tan, P., Xiong, H., Kumar, V.: *Generalizing the Notion of Support.* In Proc. 10th Int. Conf. on Knowledge Discovery and Data Mining (KDD '04), pp. 689–694.
9. Xiong, H., Tan, P., Kumar, V.: *Mining Strong Affinity Association Patterns in Data Sets with Skewed Support Distribution.* In Proc. Int. Conf. on Data Mining (ICDM'03), pp. 387–394.
10. Zaki, M., Parthasarathy, S., Ogihara, M., Li, W.: *New Algorithms for Fast Discovery of Association Rules.* In Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining (KDD '97), pp. 283–296.