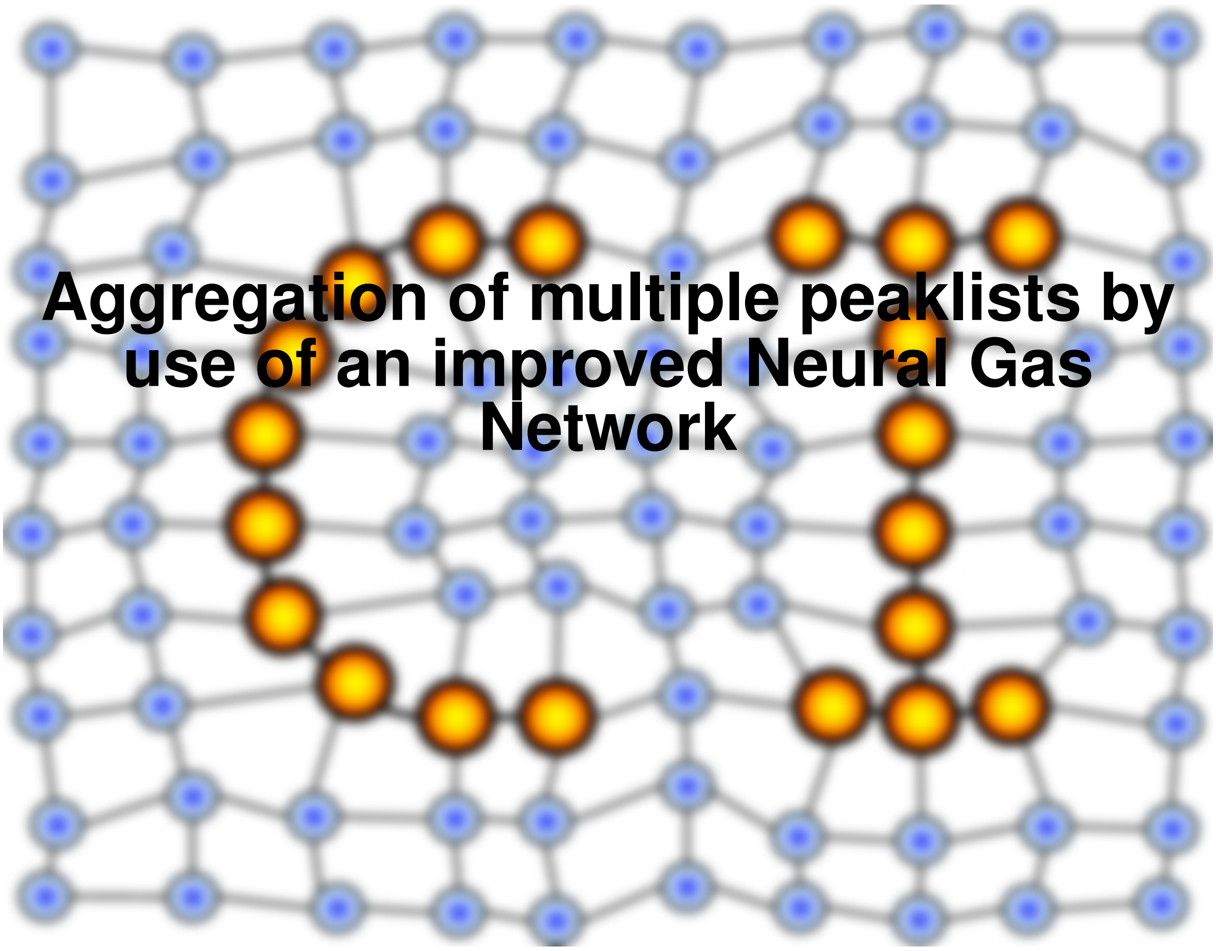
MACHINE LEARNING REPORTS

Aggregation of multiple peaklists by use of an improved Neural Gas Network



Report 02/2007

Frank-Michael Schleif[1] Alexander Hasenfuss[2] and Thomas Villmann[1]

1- University Leipzig, Dept. of Medicine, 04107, Leipzig,Germany
{schleif,villmann}@informatik.uni-leipzig.de
2- TU-Clausthal, Dept. of Math. & C.S., 38678 Clausthal-Zellerfeld,Germany
{hasenfuss}@in.tu-clausthal.de

- submitted 6th November 2007, published 24th November 2007 -

Machine Learning Reports,Research group on Computational Intelligence,
http://www.uni-leipzig.de/˜compint

Abstract

Spectra are typically interpreted by means of their peak lists. Thereby the generation of a peak list for a single spectrum is done using a peak picking algorithm. Peak lists obtained by multiple spectra which have something in common, e.g. are generated from similar sample sources, have to be combined to make more complex machine learning approaches or statistical analysis applicable. Here we present an order insensitive approach to combine multiple peak lists for mass spectrometric (MS) or nuclear magnetic resonance (NMR) metabolite measurements. The approach employs a specific batch variant of a clustering algorithm. Experimental results for MS data are given.

KEYWORDS: **peaklists, peaks, clustering, magnification control, aggregation, spectra, group comparison, MALDI, NMR, deSieno, batch-NG**

1 Introduction

The analysis of biological samples is a common task in many life science disciplines. Prominent techniques to analyse such samples are given by mass spectrometry (MS) and Nuclear magnetic resonance spectroscopy (NMR). Typical fields where such techniques are applied, are the analysis of small molecules, e.g. metabolite studies or medium or larger molecules, e.g. peptides and small proteins in case of mass spectrometry [SRS+03, KHT+05, PFL+03]. Both techniques have their own specifics but the common concept of peaks or peak lists, which cover the most relevant information extracted by the measurement process. Peak lists can be considered as a compressed, information preserving encoding of the originally measured spectra. While an appropriate generation of a peak list is a complex task, here we focus on the combination of multiple peak lists. The combination of peak lists is relevant if one is interested in comparisons between multiple spectra - encoded by means of peak lists - or group comparisons e.g. cancer and control group, which contains huge sets of spectra or peak lists. The most common approach is, to generate a common peak list at the beginning of the experiment by e.g. averaging all measured spectra. Subsequently only one peak list is calculated based on the average spectrum [KHT+05, MCK+05]. Although this approach is easy and frequently sufficient, it has some important drawbacks. One the one hand side, as pointed out in [MCK+05], the averaging improves the signal to noise (S/N) ratio of the spectra and therefore makes the application of a peak picking algorithm easier, such that also small peaks can be easily picked. This approach works fine, if the spectra belong to a common set or two groups of similar size, with similar content to be analyzed. However the averaging over multiple imbalanced and non-similar data may lead to significant prune out effects in the obtained average spectrum. Hence a peak list generated on the basis of such a spectrum is loosing significant information. To overcome this problems peak lists on single groups[1] or on single spectra can be generated. This is the best way to preserve the peak information obtained by the single spectra. However a peak picking on single spectra reveals problems with respect to S/N leading to a more complex peak picking. Further, peak lists obtained from spectra of the same sample show differences. Hence an approach to combine peak lists with some information overlap is needed.

A possible solution of the above outlines problems would be the application of a clustering algorithm on the peak lists which combines peaks which are close to each other. Here it is necessary that the clustering algorithm should be able to generate order independent clusterings such that small changes on the measurement region will not strongly interfere to the final peak lists. Such a modification may e.g. happen if the mass range on a MS measurement is trimmed such that parts of the spectra are removed. This, of course, will remove peaks from the peak lists and a subsequent clustering should not completely dump with respect to the non-pruned spectrum. Another point is the number of peak lists, which have to be combined. This number may be quite high e.g. multiple thousand peak lists with at least 100 peaks per peak lists such that the number of peaks or data points considered in the clustering easily become a million. Hence a quick optimization is needed, whose complexity scales favorable with the number of peaks or data points. Further, the number of cluster is not

[1] This may also result in problems because multiple labellings of the data may have a strong impact on the final *common* peak lists.

known beforehand but only a raw guess can be made. Generic clustering algorithms tend to represent cluster with large densities and to suppress or ignore clusters with only few datapoints. This challenge has also to account for by an appropriate peak list combination algorithm.

The report is organized as follows. First, an introduction to the neural gas algorithm as an appropriate clustering algorithm is given and its batch variant is explained in more details. Subsequently some remarks on magnification control for batch-NG and an heuristic of deSieno is explained which gives a better information preserved encoding of the peak lists such that also peaks with only few representants are still sufficiently presented. In the experimental section the approach is explained for MS data in comparison to a standard averaging approach.

2 Clustering by Batch Neural Gas

Neural gas is an unsupervised prototype based vector quantization algorithm. It maps data vectors $\mathbf{v}$ from a (possibly high-dimensional) data manifold $V \subseteq \mathbb{R}^d$ onto a set A of neurons i formally written as $\Psi_{V \to A} : V \to A$. Inputs are denoted by $\mathbf{v}$ and $V \subseteq \mathbb{R}^{D_V}$ is a finite set of inputs $\mathbf{v}$. Neural Gas (NG) uses a fixed number of prototypes (weight vectors, codebook vectors). Let $\mathbf{W} = \{\mathbf{w_r}\}$ be the set of all codebook vectors. The step of vector quantization is implemented by the map Ψ as a winner-take-all rule, i.e. a stimulus vector $\mathbf{v} \in V$ is mapped onto that neuron $\mathbf{s} \in A$ the pointer $\mathbf{w}_s$ of which is closest to the presented vector $\mathbf{v}$,

$$\Psi_{V \to \mathcal{A}} : \mathbf{v} \mapsto \mathbf{s}(\mathbf{v}) = \operatorname*{argmin}_{\mathbf{r} \in A} d(\mathbf{v}, \mathbf{w_r}) \tag{1}$$

with $d(\mathbf{v}, \mathbf{w})$ being an arbitrary distance measure, usually the squared euclidean metric. The neuron $\mathbf{s}$ is called winner or best matching unit. The subset of the input space $\Omega_\mathbf{r} = \{\mathbf{v} \in V : \mathbf{r} = \Psi_{V \to A}(\mathbf{v})\}$, which is mapped to a particular neuron $\mathbf{r}$ according to (1), forms the (masked) receptive field of that neuron. Standard NG training adapts the prototypes to represent the data as accurately as possible.

During the adaptation process a sequence of data points $\mathbf{v} \in \mathcal{V}$ is presented to the map with respect to the data distribution $P(\mathcal{D})$. Each time the current most proximate neuron s according to (1) is determined. The vector $\mathbf{w}_s$ as well as all vectors $\mathbf{w}_i$ of neurons in the neighborhood of $\mathbf{w}_s$ are shifted towards $\mathbf{v}$, according to

$$\triangle \mathbf{w}_i = -\epsilon h_\sigma(\mathbf{v}, \mathbf{W}, i) \frac{\partial d(\mathbf{v}, \mathbf{w}_i)}{\partial \mathbf{w}_i}. \tag{2}$$

The property of “being in the neighborhood of $\mathbf{w}_s$” is captured by the neighborhood function

$$h_\sigma(\mathbf{v}, \mathbf{W}, i) = \exp\left(-\frac{k_i(\mathbf{v}, \mathbf{W})}{\sigma}\right), \tag{3}$$

with the rank function

$$k_i(\mathbf{v}, \mathbf{W}) = \sum_j \theta\left(d(\mathbf{v}, \mathbf{w}_i) - d(\mathbf{v}, \mathbf{w}_j)\right) \tag{4}$$

counting the number of pointers $\mathbf{w}_j$ for which the relation $\|\mathbf{v} - \mathbf{w}_j\| < \|\mathbf{v} - \mathbf{w}_i\|$ is valid [MBS93]. $\theta(x)$ is the Heaviside-function. It should be mentioned that the neighborhood function is evaluated in the input space. The adaptation rule for the weight vectors follows in average a potential dynamic according to the potential function [MBS93]:

$$E_{NG} = \frac{1}{2C(\sigma)} \sum_j \int P(\mathbf{v})\, h_\sigma(\mathbf{v}, \mathbf{W}, j)\, d\left(\mathbf{v}, \mathbf{w}_j\right) d\mathbf{v} \tag{5}$$

with $C(\sigma)$ being a constant. It will be dropped in the following. It was shown in many applications that the NG shows a robust behavior together with a high precision of learning. Recently a batch variant of NG has been proposed in [CHHV06] which gives similar results, whereby for each update step of the prototypes all data points are considered into one step. This variant of NG is very efficient [CHHV06]. It can be extended to an information optimal variant by means of a magnification control scheme [HHV07]. This approach gives an improved coding also for cluster sets with a small number of items by magnification control, but is quite complex due to multiple density estimations. Here we consider a one dimensional problem (only mass positions) and a simple but effective alternative: the rule of deSieno [DeS88], will be investigated which approximates the magnification control effect for low dimensional data in the light of information optimal coding. The corresponding extension for NG is presented in the next section.

2.1 Magnification Control for Batch Neural Gas

We now briefly review the concepts of magnification control for Batch NG as given in [HHV07]. A characteristic property of vector quantizers consists in a selective magnification of regions of interest. This corresponds to a specific connection between the density of prototypes and stimuli. Usually, regions with high data density attract more prototypes than regions which are only sparsely covered by the data. An information theoretic optimum magnification factor corresponds to an exact adjustment of the prototypes according to the underlying data distribution i.e. $\alpha = 1$. That means, the amount of data is the same for the receptive field of every prototype. In this case, the information, which is conserved substituting the points in a receptive field by its prototypes, is maximized. For a variety of popular alternatives, however, the magnification follows a power law with exponent different from one.Popular methods include local learning, where the learning rate of the training algorithm is adjusted according to the local data density; winner relaxing strategies where the learning rate of the winner is enlarged by an additional correction to achieve optimum information transfer; and convex and concave learning where an exponent is added to the adaptation vector of the prototypes into the direction of the actual data point. In all cases, magnification control changes the learning scheme and allows to achieve a magnification factor one or beyond. An explicit control is particularly interesting for application areas where rare events should be suppressed or, contrarily, emphasized. A magnification factor α larger or smaller than one, respectively, allows to achieve this goal. Explicit magnification control has proven beneficial in several tasks in robotics and image inspection. In this paper the task of peak lists combining is a 1-D problem and a magnification for data space regions with sparse distributed data points (e.g. rare but relevant peaks) may be desired.

The magnification factor of online NG is $\alpha = D/(D+2)$ [MBS93], D being the intrinsic (Hausdorff) dimension of the data manifold of stimuli. Thus, it is different from $\alpha = 1$ in general. It approaches 1 only for very large intrinsic dimensionality, which is usually not the case. The magnification factor can be controlled using e.g. local learning, as already mentioned above. Local learning changes the learning rate by a factor depending on the local data density [Vil00]. It constitutes an intuitive learning scheme which is plausible from a biological point of view. We will focus on the local learning method in the following using Batch-NG. For Batch-NG the same magnification factor as for the online NG: $D = D/(D+2)$ can be observed [HHV07]. Here, we review magnification control for batch NG by including local learning into the update formulas. The link becomes possible because local learning can be related to a modified cost function which can be optimized in the batch mode. Intuitive update formulas arise where the new prototype locations are determined as the average of the data points weighted according to the rank and the local data density. As for standard batch NG, one can prove the convergence of batch optimization of this altered cost function.

For low intrinsic dimensionality D, which is often the case in concrete settings, the magnification factor is considerably smaller than 1. This has the consequence that regions of the input space with low data density are emphasized. Local learning extends the learning rate by a factor which depends on the local data density:

$$\delta w_i = \epsilon_0 \cdot P(w_{s(v_j)})^m \cdot h_\lambda(k_i(v_j, \mathbf{W})) \cdot (v_j - w_i)$$

where $\epsilon > 0$ is the learning rate and $s(v_j)$ is the winner index for stimulus v_j. P is the data density, $m > 0$ is a constant which controls the magnification exponent. The factor $P(w_{s(v_i)})^m$ vanishes for $m = 0$ leading to standard NG. For this online learning rule, the power law $p(w_i) \approx P(w_i)^\alpha$ results where

$$\alpha' = (m+1) \cdot \alpha = (m+1) \cdot D/(D+2)$$

as shown in [Vil00]. The information theoretic optimum factor is obtained for $m = 2/D$. Larger values emphasize input regions with high density, whereas smaller values focus on regions with rare stimuli. To apply the learning rule, the distribution P as well as the effective data dimensionality D have to be estimated from the data (using e.g. Parzen windows resp. the box counting dimension). Here, we consider the similar learning rule

$$\delta w_i = \epsilon_0 \cdot P(v_j)^m \cdot h_\lambda(k_i(v_j, \mathbf{W})) \cdot (v_j - w_i)$$

where the local density of the location of the stimulus is taken instead of the winner. The average of this learning rule can be formulated as an integral

$$< \delta w_i > \approx \int P(v)^m \cdot h_\lambda(k_i(v_j, \mathbf{W})) \cdot (v_j - w_i) \cdot P(v) dv$$

In the limit of a continuum of prototypes, $w_{s(v_i)} = v$ holds. Thus, this average update yields exactly the same result as the original one proposed in [Vil00]. Since the magnification factor of local learning has been derived under the assumption of a continuum of prototypes with $w_{s(v_i)} = v$, the same magnification factor $(m+1) \cdot \alpha'$ results for this altered learning rule. It has the benefit that it constitutes a stochastic gradient descent of the cost function

$$E_m(\mathbf{W}) = \frac{1}{2C(\lambda)} \sum_{i=1}^{n} \int P(v)^m \cdot h_\lambda(k_i(v, \mathbf{W})) \times ||v - w_i||^2 \cdot P(v) dv$$

as shown in [HHV07].

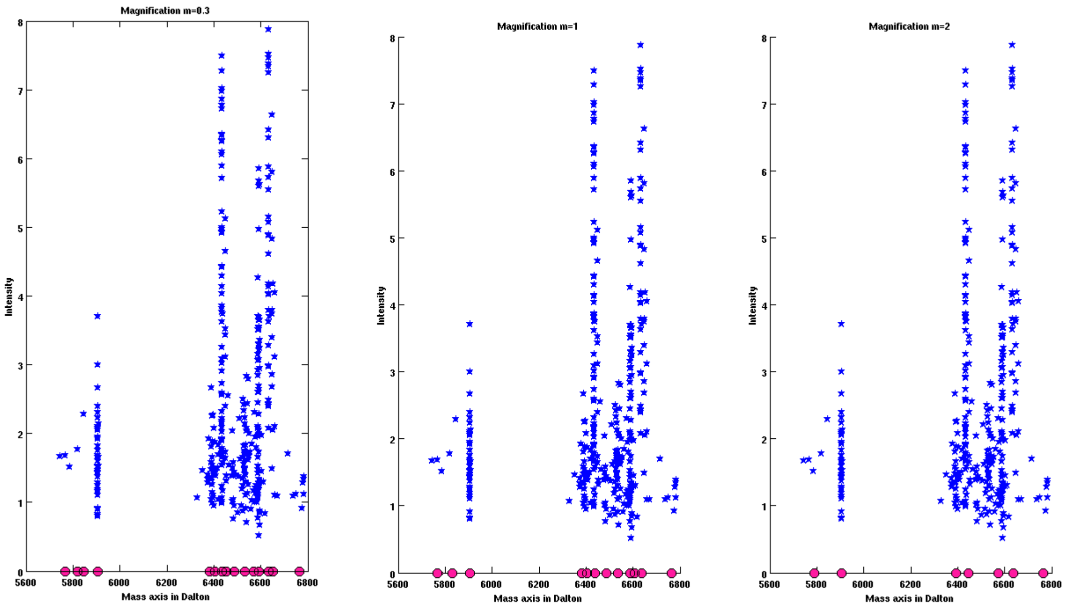


Figure 1: Three plots of clustered peak data. Clusterings are generated by Batch-NG with magnification control using different values for the magnification factor m. Data are shown as blue $\star$ and prototypes as red $\circ$ the x-axis shows the mass position of the peaks in Da whereas the y-axis gives intensity values for the corresponding peaks mapped to the respective mass position. All plot show a specific region showing the effect of magnification by means of information optimal coding of prototypes. The left plot is obtained with a magnification factor $m = 0.3$ which is close to the regular NG algorithm with $m = 0$, subsequent plots have increasing magnification factors of $m = 1.0$ and $m = 2.0$ respectively. One can clearly observe that the increasing value of m leads to a change of the number of prototypes spend for a cluster such that also regions with lower data density are reliable representable.

Thus, learning schemes which optimize the cost function $E_m(\mathbf{W})$ yield a map formation with magnification factor a'. As further pointed out in [HHV07] the cost function with magnification control for batch NG becomes

$$E_m(\mathbf{W}, K) = \frac{1}{2C(\lambda)} \sum_{i=1}^{n} \sum_{j=1}^{p} h_\lambda(k_{ij}) \cdot ||v - w_i||^2 \cdot P(v_j)^m$$

with hidden variables k_{ij} as in the original batch NG. Thus, a fast batch adaptation scheme is offered with magnification coefficient $(m+1) \cdot D/(D+2)$ which can explicitly be controlled by the quantity m. As beforehand, the local data density $P(v_j)$ has to be estimated e.g. using Parzen windows. The intrinsic data dimensionality D can be estimated using e.g. a Grassberger-Procaccia analysis such that a value m which yields optimum information transfer can be determined. The effect of magnification control in the analysis of peaklists is depicted in Figure 1.

2.2 Extended Neural Gas by the DeSieno Rule

Vector quantization distributes prototypes at representative positions of the data space, approximating the data density. Ideally every neuron should win the competition for an input with the same probability. However popular VQ schemes such as k-means do not find such an allocation, but tend to overrepresent regions with high data density, while ignoring regions with rare examples. Subsequently a simple approach presented by deSieno [DeS88] is reviewed. Magnification describes the relation of the input density $P(w)$ and the neuron density $p(w)$. This relation is usually expressed by a power law $P(w) \approx p(w)^{\alpha}$. The magnification factor for standard Neural Gas or k-Means is $D/(D+2)$ and hence for 1 dimensional data just $1/3$. This low factor indicates thats rare samples are potentially underrepresented by the algorithm. A magnification factor of 1 corresponds to a perfect match of prototype allocation and data distribution. Different methods have been proposed to control the magnification factor of an algorithm, thereby the conscience approach of deSieno has been found to be promising with respect to learning time and magnification efficiency. Frequent winners get a penalty whereas rare winners are boosted. This approach has not been applied to alternative VQ schemes such as k-means or batch SOM. Here we show the effectiveness of conscience learning for (batch) neural-gas clustering.

Each neuron $\mathbf{w_i}$ is equipped with a conscious term b_i depending on how often it has won in competitions. The conscious term b_i is subtracted from the distances in the rank determination (4) such that a bonus or penalty with respect to the winner frequency is available with:

$$b_i = C \cdot \left(\frac{1}{N} - p_i \right)$$

and

$$p_i' = p_i + B(y_i - p_{\text{iold}}) \text{ with } 0 < B << 1$$

Thereby $p_i \in \mathbb{R}$ is a winner count for the neuron $\mathbf{w_i}$, B a small learning rate constant for the winner count update usually $B = 0.0001$ and C the bias constant provided by deSieno. C is a constant which is related to the distance of a data point effecting the solution. The value y_i is the winner frequency of each prototype.

The conscience increases for the winner and decreases for all other neurons. For mass spectrometry data the user defined constant C has been fixed to $C = 10000$ which is necessary due to the unnormalized data space of mass positions with masses in the range of $1 - 10kDa$ and B was chosen as $B = 0.0001$. Updates of p_i are made after each complete run of the batch Neural Gas such that all samples have been considered exactly one time. Frequent losers get a huge bonus subtracted from their distance, frequent winners get a smaller bonus or even a penalty. Thereby the winner counts are evaluated for all prototypes in the Neural Gas and normalized with respect to the number of neurons. After a careful long training every neuron wins about the same number of training inputs and has about the same probability of winning. Here the de Sieno approach is used only, to improve the current behavior of the standard batch Neural Gas and long runs are avoided, further a prototype is a proxy for a peak. Items matched to the peak, or which are in its receptive field, have to be in close proximity. Due to this constraint an equal winning probability is not desirable for all prototypes. However even under this limitation an improvement of the magnification can be observed compared to the standard approach and the representation of sparse data regions is improved.

3 Peak aggregation by Batch Neural Gas

The full approach used to combine multiple peak lists comprises the following steps:

1. Calculate single peak lists and an average peak list by analyzing the average spectrum

2. Remove very common peaks which are already represented by peaks identified on the average spectrum

3. On the remaining number of peaks the peak aggregation using Batch NG is applied

4. The obtained reference peaks are analyzed with respect to user constraints

5. The process of peak aggregation is repeated multiple times (5)

6. The final peak list is reported and merged with the initial average peak list.

The calculation of the peak lists on an average spectrum is presented in [KHT+05] and [MCK+05]. Thereby common peaks are peaks or in this case mass positions which are already detected on the average spectrum. The average peak list is taken as an initial reference list and all peaks are matched with respect to this list. Peaks which can be sufficiently represented by this list (e.g. within a small tolerance in ppm) are removed from the set of peaks. The remaining peaks are used for batch-NG. Unrepresented peaks R_p are processed by Batch-Neural-Gas with Magnification control either in the variant of [HHV07] or using the deSieno approach as presented above. The user can define a minimal cluster size M_c, this value is used to determine a initial guess for the number of prototypes N_p needed in Batch Neural Gas in accordance to:

$$N_p = R_p/M_c;$$

if the number of clusters exceeds a predefined threshold e.g. 500 we limit the N_p to this value. The Batch Neural Gas algorithm is applied with at least 1 prototype. After the clustering the receptive fields are checked and the set of unrepresented items is reduced by the peaks which are in close proximity of a prototype by means of a minimal tolerance, such as a peak shift in PPM. Subsequently the obtained peak list is analyzed with respect to the minimal cluster size such that underloaded prototypes are removed, also the clarity of the aggregated peak such that only one peak of a single spectrum is mapped to the receptive field of the prototype is checked. In a post processing step prototypes can be merged which maybe to close to each other. This process is iterated multiple times on the remaining unrepresented peaks until all peaks are sufficiently represented or a upper number of iterations is reached. Finally only those items are not represented which could not be assigned to a prototype due the the PPM constrained or which lead to underloaded prototypes. In the last step the obtained peak list is combined with the initial peak list from the average spectrum and again all peak positions are analyzed with respect to the above mentioned user constraints. The final summarized list is reported as the final peak list.

4 Experimental results

Here we present initial results for peak list aggregation using the presented approach in comparison to a standard variant as given in [KHT$^+$05]. Thereby we explicitly include the approach of magnification control by means of magnification control for batch NG or the simpler approach of deSieno [DeS88] extended for batch neural gas as shown above. The effectiveness of both approaches on multiple data sets was already shown in [HHV07] and [DeS88], therefore we are focusing on the application of the methods in the context of peak list aggregation. In the considered scenario the peaks constitute a one dimensional data space of mass positions obtained from peak lists of multiple spectra. They aim is to obtain a common peak list, which is representative for typical peaks observed in the data. This task has to be realized by a clustering approach - in this case batch NG, which allows an order independent generation of the aggregated peak list. For the considered data it can be expected that some mass ranges are dense filled with peaks and some other are sparse. This is the motivation to incorporate magnification control to get an improved, ideally information optimum coding of the mass positions by means of prototype locations, forming the final aggregated peak list. It can be expected that magnification control improves the convergence of the aggregation procedure and improves the representation properties, such that also peaks which are rare but sufficiently common are still represented and dense regions are not over represented by a large number of prototypes.

In the first experiment we consider a synthetic data set of two classes of spiked and non-spiked proteom data measured by MALDI-MS with 25 spectra for each class. For each spectrum a peak list is generated. A common peak list should be determined. For the standard approach this common list is obtained by averaging all present spectra and the application of a peak picking on the mean spectrum. Due to the averaging the average spectrum shows a better signal to noise (s/n) ratio such that only peaks with a $S/N \geq 5$ are kept. The data are depicted in a (top view) in Figure 2. The obtained aggregated peak list of the standard approach with respect to a batch-NG variant without magnification control leads to exactly the same number of 41 peaks, which is a random effect. However, a closer inspection shows that the lists are in fact not identical. For the single peak list approach using batch NG three peaks in the lower mass range $1707Da, 1773Da, 1954Da$ are listed which are not part of the standard peak list but are correct detected. For the standard list three optional peaks $2721Da, 2744Da, 2764Da$ are detected which have very low intensities and are not detected in the batch approach because there S/N ratio was to bad such that they have been screened out in advance. These results show that the batch-NG can be successfully applied for this task and is able to generate an aggregated list of peaks.

In a subsequent analysis the batch-NG with magnification control and the batch-NG with the deSieno rule where applied. In both cases we obtain again very similar peak lists to the already obtained lists. For the batch-NG with magnification we observed quite long runtimes which can be related back to the integrated density estimations needed for magnification control, which has to be calculated once at the beginning of the batch NG algorithm. As pointed out in the iterative aggregation algorithm (IAA) the used aggregation method is applied multiple times, hence this additional computational effort is a not neglectable. Considering the runtime of the single NG run, without the costs for density estimations, the NG achieved a fast convergence and the obtained lists did already fit quite well the additional constraints stated in the IAA. Hence, a

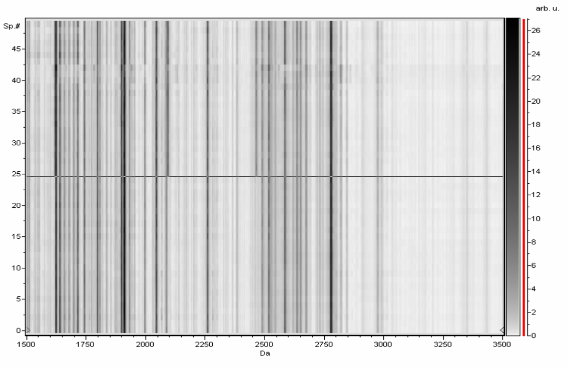


Figure 2: Gel view of synthetic data of proteom spectra. The data set consists of 25 spiked and 25 non spiked spectra.

smaller number of iterations was needed to obtain a sufficient representation of the peaks by means of an aggregated peak list. Nevertheless the computational effort was still quite high, due to the density estimation. Considering the approach of deSieno for batch-NG the following results were obtained. The peak lists generated in a single run were in general less perfect with respect to information optimal coding than those obtained by batch-NG with magnification control but in average better than the peak lists without any magnification control method. This again improved the convergence of the overall IAA method. The finally obtained peak list was again similar to the already obtained lists using batch-NG with magnification control but the procedure was faster also with accounting of the additional effort of the deSieno rule.

5 Conclusions

A method for the combination of multiple peak lists has been presented. This task is relevant in multiple fields where group comparisons are made. Here a common peak list is necessary to make more complex data analysis approaches applicable which typically rely on a feature matrix.

The presented approach gives an efficient and information optimal way to aggregate multiple peak lists. Thereby multiple constraints such as minimal occurrence frequency of a peak or peak shift tolerance can be dealed with. Further the chosen variant of Batch-NG allows a fast less order independent peak aggregation in a newton optimization scheme.

To obtain an information optimal representation of the peaks in the codebook model and to improve the convergence performance of the aggregation method, two kinds of magnification control have been tried. Thereby the best coding was obtained by use of the magnification control based batch-NG, followed by batch-NG with deSieno. The most computational effective approach is given by the standard approach. Thereby all spectra are average and the final peak list is obtained by considering the peak picking results on the average spectrum. This however is not optimal in case of multiple very dissimilar classes and hence a more generic approach is desirable. Taking this fact into account a single peak picking approach is needed and thereby the computational most effective, by means of runtime and optimal coding, approach was found to the one of batch-NG using the deSieno rule. This is due to the fact that the necessary modification of the batch-NG scheme are relatively simple but the effect on the magnification is still sufficient to improve the overall performance of the IAA scheme. Hence the following suggestions can be made. If only two class of approximately the same number of spectra is analyzed, the average spectrum approach is the most effective one. In case of multiple classes and/or very unbalanced spectra sets, a single peak picking approach with an IAA aggregation scheme should be used. Thereby the incorporation by means of the deSieno rule is most effective with respect to runtime and a reliable good information coding. For information optimal coding in an IAA scheme the batch-NG with magnification control is preferable on the drawback of a higher computational effort.

*A*cknowledgement: This work was supported by the BMBF in the project MetaSTEM. The authors would like to thank Marc Gerhard, Bruker Daltonik GmbH, Bremen, Germany for helpful discussions and support in implementing the approach in the Spectra processing software ClinProTools.

References

[CHHV06] COTTRELL, M.; HAMMER, B.; HASENFUSS, A.; VILLMANN, T.: Batch and median neural gas. In: *Neural Networks* 19 (2006), Nr. 6-7, S. 762–771

[DeS88] DESIENO, Duane: Adding a conscience to competitive learning. In: *Proc. ICNN'88, International Conference on Neural Networks*. Piscataway, NJ : IEEE Service Center, 1988, S. 117–124

[HHV07] HAMMER, B.; HASENFUSS, A.; VILLMANN, T.: Magnification control for Batch Neural Gas. In: *Neurocomputing* 70 (2007), Nr. 7-9, S. 1225–1234

[KHT+05] KETTERLINUS, R.; HSIEH, S-Y.; TENG, S-H.; LEE, H.; PUSCH, W.: Fishing for biomarkers: analyzing mass spectrometry data with the new ClinPro-Tools software. In: *Bio techniques* 38 (2005), Nr. 6, S. 37–40

[MBS93] MARTINETZ, Thomas M.; BERKOVICH, Stanislav G.; SCHULTEN, Klaus J.: 'Neural-Gas' Network for Vector Quantization and its Application to Time-Series Prediction. In: *IEEE Trans. on Neural Networks* 4 (1993), Nr. 4, S. 558–569

[MCK+05] MORRIS, Jeffrey S.; COOMBES, Kevin R.; KOOMEN, John; BAGGERLY, Keith A.; KOBAYASHI, Ruji: Feature extraction and quantification for mass spectrometry in biomedical applications using the mean spectrum. In: *Bioinformatics* 21 (2005), Nr. 9, S. 1764–1775

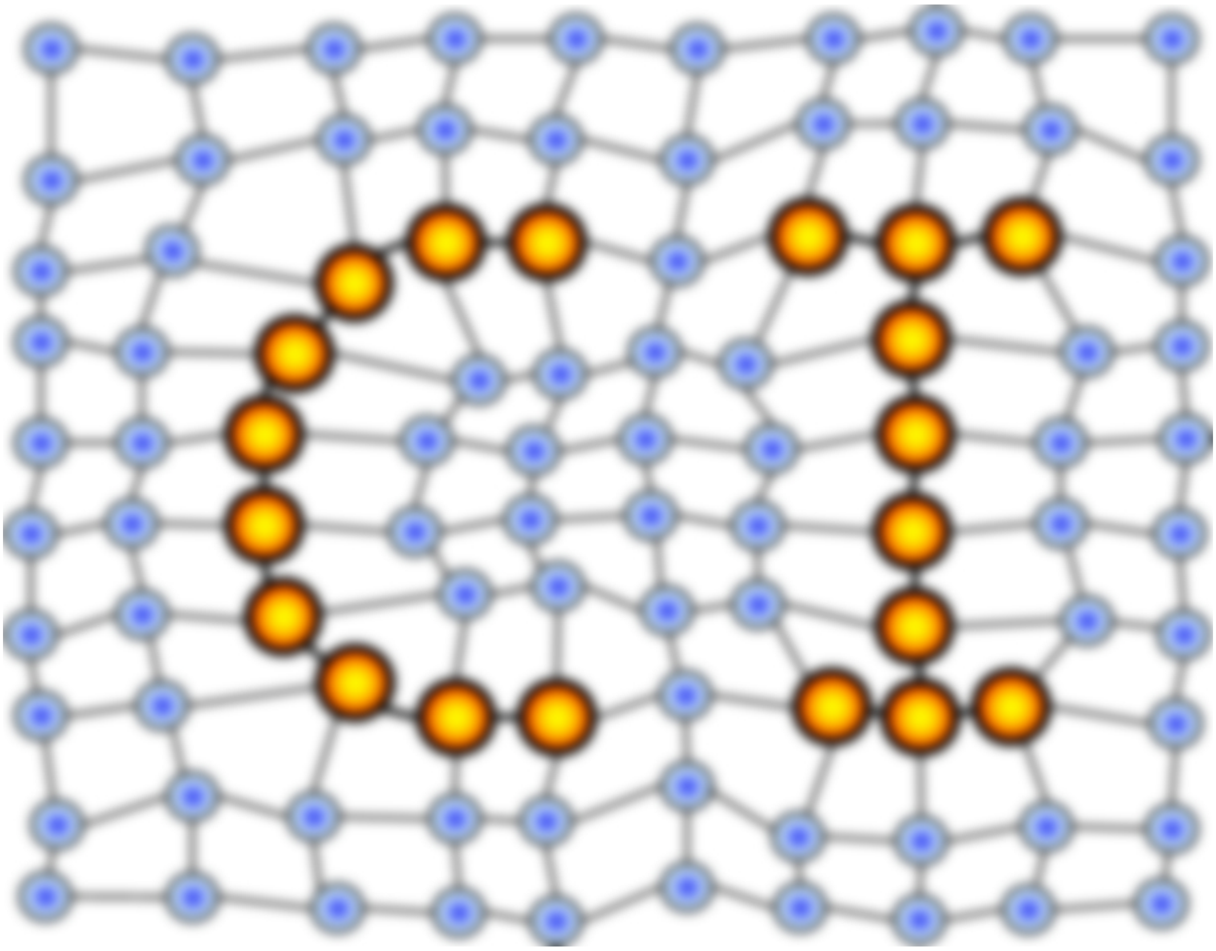
[PFL+03] PUSCH, W.; FLOCCO, M.; LEUNG, S.M.; THIELE, H.; KOSTRZEWA, M.: Mass spectrometry-based clinical proteomics. In: *Pharmacogenomics* 4 (2003), S. 463–476

[SRS+03] SUCKAU, D.; RESEMANN, A.; SCHUERENBERG, M.; HUFNAGEL, P.; FRANZEN, J.; HOLLE, A.: A novel MALDI LIFT-TOF/TOF mass spectromety for proteomics. In: *Anal. Bioanal. Chem* 376 (2003), S. 952–965

[Vil00] VILLMANN, Thomas: Controlling Strategies for the Magnification Factor in the Neural Gas Network. In: *Neural Network World* 10 (2000), Nr. 4, S. 739–750

Machine Learning Reports

Report 02/2007



Impressum

Machine Learning Reports ISSN: 1865-3960

▽ Publisher/Editors

PD. Dr. rer. nat. Thomas Villmann & Dr. rer. nat. Frank-Michael Schleif
Medical Department, University of Leipzig
Karl-Tauchnitz 25, D-04107 Leipzig, Germany •
http://www.uni-leipzig.de/˜compint

▽ Acknowledgments

We would like to thank the reviewers for their time and patience.

Machine Learning Reports,Research group on Computational Intelligence,
http://www.uni-leipzig.de/˜compint