Published in: "**An electronic text book: Electron microscopy in Life Science**",
3D-EM Network of Excellence, Editors: A. Verkley and E. Orlova (2009).

Multivariate Statistical Analysis in Single Particle (Cryo) Electron Microscopy*

Marin van Heel[1], Rodrigo Portugal[1], and Michael Schatz[2]

[1] Imperial College London, Biochemistry Building,
London SW7 2AZ, England

[2] Image Science Software GmbH, Gillweg 3
D - 14193 Berlin, Germany

Key words: Single particles, cryo-EM, MSA, Multivariate Statistical Analysis, correspondence analysis, parallelisation, symmetry analysis, sample heterogeneity, Correlations, Euclidian Distances, Chi-square distances, Modulation metrics.

Abstract

Biology is a challenging and complicated mess. Understanding this challenging complexity is the realm of the biological sciences: trying to make sense of the massive, messy data in terms of discovering patterns and uncovering underlying general rules. Among the most powerful mathematical tools for organising and helping to structure complex, heterogeneous and noisy data are the tools provided by the family of multivariate statistical analysis (MSA) approaches. In electron microscopy (EM), MSA was first introduced to in 1980, to help sort out different views of macromolecules in a micrograph. After almost 30 years of continuous use and developments, new MSA applications are still being proposed regularly. The speed of general computing has increased dramatically in the three decades since the first use of MSA in electron microscopy. However, we have seen an even more rapid increase in the size and the complexity of the EM data sets we wish to analyse. Therefore, the speed of the MSA computations has become a very serious bottleneck limiting its use. The recent parallelisation of MSA programs open whole new possibilities by making the process to run orders of magnitudes faster exploiting the capacity of hundreds of CPUs simultaneously. The purpose of this paper is to not only intuitively explain the basic principles of multivariate statistical eigenvector-eigenvalue data compression to the novice in the field, but also to provide the more experienced researcher in structural biology with the formulas associated with the various MSA approaches.

*** This paper is dedicated to the memory of Jean-Pierre Bretaudière (see appendix)**

1. Introduction

The electron microscope instrument, invented primarily by Ernst Ruska in the nineteen thirties, became a routine scientific instrument during the nineteen fifties and sixties. With the gradual development of the appropriate specimen-preparation technique it proved an invaluable tool for visualising biological complexes. For example, ribosomes, originally named "Palade particles" were first discovered in the nineteen fifties in electron microscopical images [Palade 1955]. In the nineteen sixties and seventies, the early days of single-particle electron microscopy (EM), the main specimen preparation technique used for investigating the structure of biological macromolecules was the negative stain technique [van Bruggen 1962a,b]. In those days the standard way of interpreting the structures was to come up with an intuitive three-dimensional arrangement of subunits that would fit with the observed (noisy) molecular images.

Around 1970, in a number of ground-breaking publications, the idea was introduced of using images of highly symmetric protein assemblies such as helical assemblies or icosahedral viral capsids [DeRosier & Klug 1968; Crowther 1971] to actually calculate the three-dimensional ("3D") structures of these assemblies. The images of these highly symmetric assemblies can often be averaged in three dimensions without extensive pre-processing of the associated original images. Averaging the many unit cells of a two-dimensional crystal, in combination with tilting of the sample holder gave the very first 3D structure of a membrane protein [Unwin & Henderson 1975]. Electron tomography of single particles had been proposed by Hoppe and his co-workers [Hoppe 1974] however, due to the radiation-sensitivity of biological macromolecules to electrons, it is not feasible to expose a biological molecule to the, say, one hundred times higher dose required to reveal the 3D structure from a one hundred different projection images.

For all other types of irregular complexes, no methods were available for investigating their 3D structures. The vast majority of the publications of those days thus were based on the visual recognition of specific molecular views and their interpretation in terms of the three-dimensional of the macromolecules. For example a large literature body existed on the 3D structure of the ribosome based antibody labelling experiments [Kastner 1981, Lake 1976]. The problem with the visual interpretation of two-dimensional molecular images was obviously the lack of objectivity and reproducibility in the analyses. Based on essentially the same data, entirely different 3D models for the ribosome had been proposed [Kastner 1981]. There clearly was a need for more objective methods for dealing with two-dimensional images of three-dimensional biological macromolecules.

Against this background Marin van Heel and Joachim Frank started a joint project in 1979 to allow for objectively recognising specific views in a negative stain preparation so as to be able to average similar images prior to further processing and interpretation. Averaging is a necessity in single–particle processing in order to improve upon the very poor signal-to-noise ratios ("SNR") of direct raw electron images. Averaging similar images from a mixed population of images, however, only makes sense if that averaging is based on a coherent strategy to decide which images are sufficiently similar to warrant their averaging. We need good similarity measures between images such as correlation values or distance criteria for the purpose. Upon a suggestion of Jean-Pierre Bretaudière (see appendix) the first application of multivariate statistical analysis (MSA) emerged in the form of correspondence analysis

[van Heel & Frank 1980; van Heel & Frank 1981]. Correspondence Analysis (CA) is based on Chi-square distances [Benzécri 1973-1980; Lebart 1977-1984] which distances are excellent for positive data. In retrospect, other distances are more appropriate for use in transmission electron microscopy [Borland & van Heel 1990] but let us not jump ahead and first discuss some of the basics of MSA.

A further fundamental development must be mentioned here that greatly influenced the field of single-particle electron microscopy. In the early nineteen eighties the group around Jacques Dubochet at the EMBL in Heidelberg pioneered the vitreous-ice embedding technique for biological macromolecules [Adrian 1984]. That specimen preparation technique represented a quantum leap in structural preservation of biological specimens in the harsh vacuum of the electron microscope. With classical dry negative-stain preparation technique, the earlier standard, the molecules tended to distort and lie in just a few preferred orientations on the support film. The vitreous-ice (or better: vitreous water) technique greatly improved the applicability of quantitative single particle approaches in electron microscopy including the MSA approaches.

2. The MSA problem: hyperspaces and data clouds

An image is simply a "measurement" that can be seen as a collection of numbers, each number representing the density in one pixel of the image. For example, let us assume the images we are interested in are of the size of 300x300 pixels. This image thus consists of 300x300 = 90,000 density values, starting at the top left with the density of pixel (1,1) and ending at the lower right with density of pixel (300,300). The fact that an image is intrinsically two dimensional is not really very relevant for what follows. What is relevant is that we are trying to make sense out of a large number of comparable measurements, say 200,000 images, all of the same size with pixels arranged in the same order. Each of these measurements can be represented formally by a *vector* of numbers $\boldsymbol{F(a)}$, where **a** is an index running over all pixels that the image contains (90,000 in our case).

A "vector" of numbers can be seen as a line from the origin, and ending at one specific point of a multi-dimensional space known as *hyperspace*. There is no magic in hyperspace; it is merely a convenient way in which to represent of a measurement. A set of 200,000 images is a set of 200,000 measurements, each of which corresponds to a point in hyperspace (90,000 dimensional in our case). When trying to make sense of these 200,000 different images of molecules collected in our data set, we like think in terms of distances between these points: those that lie close together in hyperspace correspond to images that are very similar! The collection of our 200,000 points in hyperspace is called a *data cloud*. Similar images are close together in the data cloud and are separated by only a small *distance*, or, equivalently, have a high degree of *correlation* as will be discussed in the next chapter. This abstract representation of sets of measurements is applicable to any form of multidimensional data including: one-dimensional (1D) spectra, two-dimensional (2D) images, or even three-dimensional (3D) volumes.

The collection of points in hyperspace forms a data *cloud,* which cloud is an exact representation of the full data set. The entire raw data set in our example consists of 200,000 images of each 90,000 pixels or a total of 18×10^9 pixel density measurements. In the hyperspace representation this translates to a cloud of 200,000 points in a 90,000 dimensional

hyperspace: again consisting of 18 x 10^9 co-ordinates. Each co-ordinate corresponding one of the original pixel density: the hyperspace representation does not change the data in any way. This type of representation is illustrated in **Fig 1** for a data set simplified to the extreme: each image consisting of just two pixels.

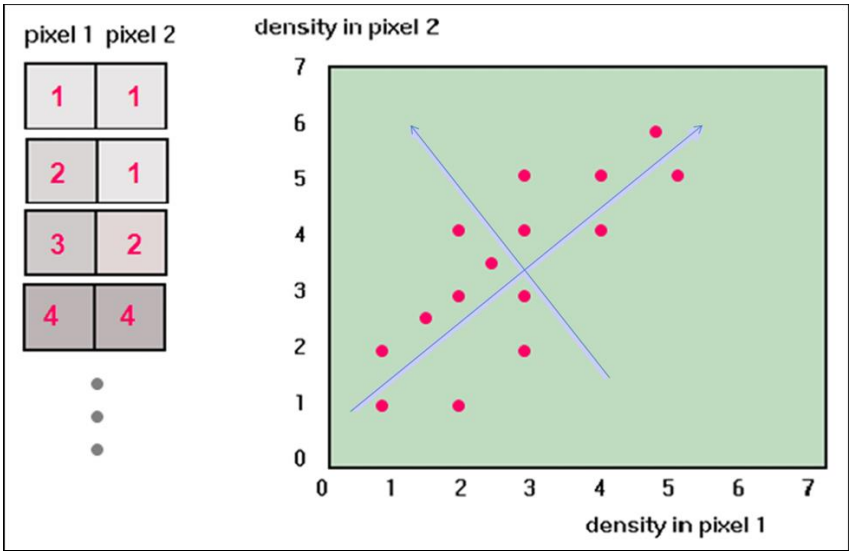

Fig 1: Hyperspace representation of an (extremely) simple set of images, each image consisting of only two pixels. Thus, two numbers completely determine each raw image in this minimalistic data set. Each image is fully represented by a single point in a two-dimensional hyperspace. Together, these points form a data "cloud". The cloud has a shape as indicated in this example. The main purpose of "MSA" approaches is to optimally adapt the co-ordinate system of this hyperspace to the shape of the cloud, as indicates by the blue arrows in this picture. The shape of the data cloud indicates that the largest source of variations in this data set is that of the densities of both pixels increasing together. That single (rotated) direction describes most of the differences between the images of the dataset.

The basic idea of the MSA approach is to optimise the orthogonal co-ordinate system of the hyperspace to best fit the shape of the cloud. We wish to rotate (and possibly shift) the co-ordinate system, such that the first axis of the rotated co-ordinate system will best correspond to the direction of the largest elongation of the data cloud. In the simplistic (two-pixel images) example of **Fig 1**, the largest elongation of the cloud is associated with the average density in the two pixels increasing together. That main direction points from the lower left of the illustration (both pixels have low density) to the top right (both pixels high density). The remaining direction is perpendicular to first one (also indicated in the illustration) but that direction describes only small modulations with respect to the main trend of the data set and may be ignored. The power of the MSA approach lies in this *data reduction*. It allows us to them concentrate on the most important trends and variations found in a complex data set and ignore all the other sources of fluctuations (which in EM usually is just *noise*). We thus enormously reduce the total amount of data into just a few main components which show the most significant variations within the data.

Concentrating on the main direction of variation of the data in the example of **Fig 1**, reduces the problem from a two-dimensional one to just a one-dimensional problem. This *reduction of dimensionality* can take dramatic proportions with real EM data sets. In the case of a data set of two-hundred thousand images of 300x300 pixels, typically some 50 orthogonal

directions suffice to describe the most significant (largest) sources of variations within the data. Each image is then no longer described by the 90,000 density values of its pixels, but rather by just its 50 co-ordinates with respect to those main directions of variations (*eigenvectors*). This represents a reduction in the dimensionality of the data by more than 3 orders of magnitude. After this data reduction, it becomes feasible to perform exhaustive comparisons between all images in the data set (compare distances for example) at a reasonable cost in terms of the necessary computing.

3. Distances and Correlations: the choice of the metric

Multivariate Statistical Analysis is all about comparing large sets of measurements and the first question to be resolved is how to compare them, that is, to decide what measure of similarity one would want to use. The concepts of distances and correlations between measurements are closely related as we will see below. Different distance and associated correlation criteria are possible depending on the *metric* one chooses to work with. We will start with simplest and most widely used metric: the classical Euclidean metric.

a) Euclidean Metrics

The classical measure of similarity between two measurements ***F(a)*** and ***G(a)*** is the *correlation* or *inner product* (also known as the *covariance*) between the two measurement vectors:

$$C_{FG} = \frac{1}{p} \sum_{a=1,p} (F(a) \cdot G(a)) \qquad (\mathbf{1})$$

The summation in this correlation between the two vectors is over all possible values of ***a***, in our case, the ***p*** pixels of each of the two images being compared. (This summation will be implicit in all further formulas; implicit may also be the normalisation by ***1/p***). Note that when ***F*** and ***G*** are the same (***F = G***), this formula yields the *variance* of the measurement: the average of the squares of the measurement:

$$C_{FF} = \frac{1}{p} \sum_{a=1,p} (F(a) \cdot F(a)) = \frac{1}{p} \sum_{a=1,p} F(a)^2 = \frac{1}{p} F_{VAR} \qquad (\mathbf{1a})$$

Closely related to the correlation value is the "Euclidean square distance" between the two measurements ***F*** and ***G***:

$$D^2_{FG} = \sum (F(a) - G(a))^2 \qquad (\mathbf{2}).$$

The relation between the correlation and the Euclidean distance between ***F*** and ***G*** becomes clear when we work out equation (**2**):

$$D_{FG}^2 = \sum (F^2(a) - 2 \cdot F(a) \cdot G(a) + G^2(a)) \qquad (\mathbf{3})$$

$$= \sum F^2(a) + \sum G^2(a) - 2 \cdot \sum F(a) \cdot G(a)$$

$$= \sum F^2(a) + \sum G^2(a) - 2 \cdot \sum F(a) \cdot G(a)$$

$$= F_{VAR} + G_{VAR} - 2 \cdot C_{FG} \qquad (\mathbf{4})$$

In other words, the Euclidean square distance between the two measurements ***F*** and ***G***, is a constant (the sum of the total variances in ***F*** and in ***G***, $\boldsymbol{F_{VAR} + G_{VAR}}$, respectively, minus twice $\boldsymbol{C_{FG}}$, the correlation between ***F*** and ***G***. Thus correlations and Euclidean distances are directly related in a simple way: the shorter the distance between the two, the higher the correlation between ***F*** and ***G***; when their distance is zero, their correlation is at its maximum. This metric is the most used metric in the context of multivariate statistics; it is namely the metric associated with Principal Components Analysis (*PCA*, see below). Although this is in general a good metric for signal processing in general, there are some disadvantages associated the with use of pure Euclidean metrics.

One disadvantage of Euclidean distances and correlations are their sensitivity to a multiplication by a constant. For example, suppose the two measurements ***F*** and ***G*** have approximately the same variance and one would then multiply one of the measurement, say ***F(a)***, with the constant value of "10". A multiplication of a measurement by such a constant does not change the information content of that measurement. However, the correlation value $\boldsymbol{C_{FG}}$ between the measurements ***F*** and ***G*** (equation (**1**)) will increase by a factor 10. The Euclidean square distance will, after this multiplication, be totally dominated by the $\boldsymbol{F_{VAR}}$ term which will then be one hundred times larger than the corresponding $\boldsymbol{G_{VAR}}$ term in (equation (**4**)).

A further problem with the Euclidean Metric (and with all other metrics discussed here) is the distorting influence that additive constants can have. Add a large constant to the measurement ***F*** and ***G***, and their correlation (equation (**1**)) and Euclidean distance (equations (**2**-**4**)) will be fully dominated by these constants, leaving just very small modulations associated with the real information content of each of ***F*** and ***G***. A standard solution to these problems in statistics is to correlate the measurements only after subtracting the average and normalising them by the standard deviation of each measurement. The correlation between ***F*** and ***G*** thus becomes:

$$C_{FG} = \sum ((F(a) - F_{AV}) / F_{SD} \cdot (G(a) - G_{AV}) / G_{SD}) \qquad (\mathbf{5})$$

$$= \sum (F'(a) - G'(a))^2 \qquad (\mathbf{5'})$$

$$D_{FG}^2 = \sum ((F(a) - F_{AV}) / F_{SD})^2 + \sum ((G(a) - G_{AV}) / G_{SD})^2$$

$$- 2 \cdot \sum ((F(a) - F_{AV}) / F_{SD} \cdot (G(a) - G_{AV}) / G_{SD})) \qquad (\mathbf{6})$$

$$= \sum F'^2(a) + \sum G'^2(a) - 2 \cdot \sum F'(a) \cdot G'(a) \qquad (\mathbf{6'})$$

This normalisation of the data is equivalent to replacing the raw measurements ***F(a)*** and ***G(a)*** by its normalised versions ***F'(a)*** and ***G'(a)***:

$$F'(a) = \quad (F(a) - F_{AV}) / F_{SD} \qquad (\mathbf{7})$$

$$G'(a) = \quad (G(a) - G_{AV}) / G_{SD} \qquad (\mathbf{8}),$$

and these substitutions introduced above thus render the Euclidean metrics correlations and distances (equations (**5**) and (**6**)) to exactly the same form as the original ones (equations (**1**) and (**2**)).

b) Pre-treatment of the data

Interestingly, it is a standard procedure to "pre-treat" EM images prior to any processing and during the various stages of the data processing and this routine is, in fact, a generalisation of this standard normalisation in statistics discussed above. During the normalisation of the molecular images [van Heel & Stöffler 1985] one first high-pass filters the raw images to remove the very low spatial frequencies. These low spatial frequencies are associated mainly with long-range fluctuations in the image density on scale of ~20 nm and above. Such long-range fluctuations are not directly related to the structural details we are interested in, and they often interfere with the alignment procedures required to bring the images in register.

The high-pass filtering is often combined with a low pass filter to remove some noise in the high-frequency range, again, trying to reduce structure-unrelated noise. These high spatial frequencies, however, although very noisy also contain the finest details one hopes to retrieve from the data. For a first 3D structure determination, it may be appropriate to suppress the high frequencies. During the 3D structural refinements, the original high frequency information in the data may later be reintroduced. The overall filtering operation is known as "band-pass" filtering.

The high-pass filtering indeed removes the very-low frequencies in the data but that also has the effect of setting the average density in the images to a zero value. In cryo-EM the predominant contribution to the image generated in the transmission electron microscope ("TEM") is phase contrast. In phase contrast, the very low frequency information in the images is fundamentally not transmitted by the phase-contrast imaging device, because that area of the back focal plane of the imaging device is associated with the image of the illuminating source or "zero order beam" [Zernike 1942a,b]. Therefore, in cryo-EM the image information is modulated around the value of zero. The average phase contrast modulations in an electron micrograph, over a sufficiently large area, are thus necessarily always essentially zero. Strictly speaking the background value in a phase-contrast image is not "zero" but rather the average density in the image, a value which is not normally experimentally available. The band-pass filtering allows one to concentrate on a range of structural details sizes that are important at the current level of processing or analysis. It has a clear relation to the standard normalization of measurements, as in equations (**7**) and (**8**), but has a much richer range of applications.

c) Chi-Square Metrics (χ^2)

As was mentioned above, the first applications of MSA techniques in electron microscopy [van Heel & Frank 1980; 1981] focused on "correspondence analysis" [Benzécri 1973; 1980; Lebart 1977; 1984] which is based on Chi-square metrics (χ^2). Chi-square distances are good for the analysis of histogram data, data that is *per definition* positive. The chi-square correlation and distance are, respectively:

$$C_{\chi FG} = \sum (F(a)/F_{AV} \cdot G(a)/G_{AV}) \qquad (\mathbf{9})$$

and

$$D^2_{\chi FG} = \sum (F(a)/F_{AV} - G(a)/G_{AV})^2 \qquad (\mathbf{10}).$$

With the chi-square metrics, the measurements ***F*** and ***G*** are normalized by the average of the measurements:

$$F'(a) = F(a)/F_{AV} \qquad (\mathbf{11})$$

$$G'(a) = G(a)/G_{AV} \qquad (\mathbf{12})$$

Substituting the normalized measurements **(11)** and **(12)** into formulas **(9)** and **(10)** brings us again back to the standard forms **(1)** and **(2)** for the correlation and distance.

Why this normalisation by the average? Suppose that of the 15 million inhabitants of Beijing, 9 million own a bicycle, and 6 million do not own a bicycle. Suppose also that we would like to compare these numbers with the numbers of cyclists and non-cyclists in Cambridge, a small university city with only 150,000 inhabitants. If the corresponding numbers for Cambridge are 90,000 bicycle owners versus 60,000 non-owners, then the chi-square distance **(10)** between these two measurements is zero, in spite of the 100 fold difference in population size between the two cities. This metric χ^2 is thus well suited for studying histogram-type of information.

Interestingly, with the χ^2-distance, the idea of subtracting the average from the measurements in already "built in", and leads to identically the same distance (**10**) as can be easily verified:

$$\begin{aligned} D^2_{\chi FG} &= \sum ((F(a)-F_{AV})/F_{AV} - (G(a)-G_{AV})/G_{AV})^2 \\ &= \sum ((F(a)/F_{AV}-1) - (G(a)/G_{AV}-1))^2 \\ &= \sum (F(a)/F_{AV} - G(a)/G_{AV})^2 \qquad (\mathbf{10a}). \end{aligned}$$

This illustrates that the χ^2-metrics are oriented towards an analysis of positive histogram data, that is, towards data were the role of the standard deviation $\boldsymbol{F_{SD}}$ and of the average $\boldsymbol{F_{AV}}$ in equation (**7**), are both covered by the average of the measurement $\boldsymbol{F_{AV}}$. Although this leads to nice properties in the representation of the data (see below) problems arise when the measurements ***F(a)*** are not histogram data but rather a non-positive signal. The normalisation by the average $\boldsymbol{F_{AV}}$ rather than by the standard deviation $\boldsymbol{F_{SD}}$ may then lead to an explosive behaviour of ***F(a)*** / $\boldsymbol{F_{AV}}$ when the average of the measurements gets close to zero.

d) Modulation Metrics

As discussed above, a disadvantage of the Euclidean distance and of the associated correlation is the sensitivity to multiplication of one of the data vectors by a constant. The χ^2-metric does not have this sensitivity to a multiplicative factor through its normalisation of the measurements by their average. In contrast to histogram data, however, in signal and image-processing the measurements need not be positive. Signals often have (or are normalized to) zero average density as discussed in the paragraph above. The χ^2-metric, when applied to signal-processing measurements is associated with a fundamental problem. This problem in the application of correspondence analysis *per se* to electron microscopical data was realised some time after its introduction in electron microscopy, and a new "modulation"-oriented metric was introduced to circumvent the problem [Borland & van Heel 1990]. With the modulation distance, one divides the measurements by their standard deviation ("SD"). The correlation value and the square distance with modulation metrics thus becomes:

$$C_{m\,FG} = \sum (F(a)/F_{SD} \cdot G(a)/G_{SD}) \qquad (\mathbf{13}),$$

and,

$$D^2_{m\,FG} = \sum (F(a)/F_{SD} - G(a)/G_{SD})^2 \qquad (\mathbf{14}).$$

With

$$F_{SD} = \sqrt{\frac{1}{p} \sum (F(a))^2} \qquad (\mathbf{15}),$$

and,

$$G_{SD} = \sqrt{\frac{1}{p} \sum (G(a))^2} \qquad (\mathbf{16}).$$

The MSA variant with modulation distances we call "modulation analysis", and this MSA technique shares the generally favourable properties of CA, yet, as is the case in PCA, allows for the processing of zero-average-density signals.

4. Matrix formulation: some basics

We have thus far considered an image (or rather any measurement) as a vector named ***F(a)*** or as a vector named ***G(a)***. We want to study large data sets of **n** different images (say: 200,000 images), each containing **p** pixels (say 90,000 pixels). For the description of such data sets we use the much more compact "matrix notation" (details to be found, for example, in Wikipedia). For completeness, we will repeat some basic (standard) matrix formulation here allowing the first time reader to remain within the notation used here. (Be aware that many different – sometimes conflicting – nomenclatures are in use in statistics).

a) The Data Matrix

In matrix notation we describe the whole data set by a single symbol, say "***X***". ***X*** stands for a rectangular array of values containing all **n** x **p** density values of the data set (say: 200000 x 90000 measured densities). The matrix ***X*** thus contains **n** rows, one for each measured image, and each row contains the **p** pixel densities of that image:

$$X = \begin{pmatrix} x_{1,p} & x_{1,2} & x_{1,3} & \dots & \dots & \dots & x_{1,p} \\ x_{2,1} & x_{2,2} & x_{2,3} & \dots & \dots & \dots & x_{2,p} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_{n,1} & x_{n,2} & x_{n,3} & \dots & \dots & \dots & x_{n,p} \end{pmatrix} \quad \begin{matrix} \\ \\ \\ \\ \\ \end{matrix} \; \text{n} \qquad \text{p} \qquad (\mathbf{17})$$

b) Correlations between matrices

This notation is much more compact than the one used above because we can, for example, multiply the matrix ***X*** with a vector ***G*** (say an image with p pixels) to yield a correlation vector ***C*** as in:

$$X \cdot G = \begin{pmatrix} x_{1,p} & x_{1,2} & x_{1,3} & \dots & \dots & \dots & x_{1,p} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_{i,1} & x_{i,2} & x_{i,3} & \dots & \dots & \dots & x_{i,p} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ x_{n,1} & x_{n,2} & x_{n,3} & \dots & \dots & \dots & x_{n,p} \end{pmatrix} \cdot \begin{pmatrix} g_1 \\ g_2 \\ \dots \\ \dots \\ \dots \\ g_p \end{pmatrix} = \begin{pmatrix} c_1 \\ \dots \\ c_i \\ \dots \\ c_n \end{pmatrix} \qquad (\mathbf{18})$$

The result of this multiplication ***X*·*G*** is a vector ***C***, which has **n** elements, and any one element of ***C***, say C_i has the form:

$$c_i \quad = \quad \sum_{a=1,p} (x_{i,a} \cdot g_a) \qquad (\mathbf{19})$$

This sum is identical to the correlation or inner product calculation presented above for the case of the Euclidean metrics (equation (**1**)). Explained in words: one here multiplies each element from row **i** of the data matrix ***X*** with the corresponding element of vector ***G*** to yield a vector ***C***, the "**n**" elements of which are the correlations (or inner products or "projections") of all the images in the data set ***X*** with the vector ***G***.

An important concept in this matrix formulation is that of the "transposed" data matrix $\boldsymbol{X}$ denoted as $\boldsymbol{X^T}$:

$$X^T = \begin{pmatrix} x_{1,1} & x_{2,1} & \cdots & \cdots & x_{n,1} \\ x_{1,2} & x_{2,2} & \cdots & \cdots & x_{n,2} \\ x_{1,3} & x_{2,3} & \cdots & \cdots & x_{n,3} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{1,p} & x_{2,p} & \cdots & \cdots & x_{n,p} \end{pmatrix} \quad \begin{matrix} \uparrow \\ \\ \\ \\ \\ \\ \\ \downarrow \end{matrix} \; \mathrm{p}$$

$$\longleftrightarrow \quad \mathrm{n} \qquad (\mathbf{20})$$

In $\boldsymbol{X^T}$, the transposed of $\boldsymbol{X}$, the columns have become the images, and the row have become what first were the columns in $\boldsymbol{X}$. Similar to the multiplication of the matrix $\boldsymbol{X}$ with the vector $\boldsymbol{G}$, discussed above, we can calculate the product between matrices, provided their dimensions match. We can multiply the $\boldsymbol{X}$ with $\boldsymbol{X^T}$ because the rows of $\boldsymbol{X}$ have the same length **p** as the columns of $\boldsymbol{X^T}$:

$$A_n \;=\; X \cdot X^T \qquad (\mathbf{21a})$$

This matrix multiplication is like the earlier one (**18**) of the (**n** x **p)** matrix X with a single vector $\boldsymbol{G}$ (of length **p**) yielding a vector $\boldsymbol{C}$ of length **n**. Since $\boldsymbol{X^T}$ is itself a (**p** x **n**) matrix the inner-product operation is here applied to each column of $\boldsymbol{X^T}$ separately, and the result thus is an (**n** x **n**) matrix $\boldsymbol{A_n}$. Note that each element of the matrix $\boldsymbol{A}$ is the inner product or co-variance between two images (measurements) of the data set $\boldsymbol{X}$. The **n** diagonal elements of $\boldsymbol{A_n}$ contain the *variance* of each of the measurements. (The variance of a measurement is the co-variance of an image with itself). The sum of these **n** diagonal elements is the total variance of the data set, that is, the sum of the variances of all images together, and it is known as the *trace* of **A**. The matrix $\boldsymbol{A}$ is famous in multivariate statistics and is called the "variance co-variance matrix". Note that we also have in the conjugate representation (see below):

$$A_p \;=\; X^T \cdot X \qquad (\mathbf{21b})$$

c) **The transposing of a product of matrixes**

Note that the matrix $\boldsymbol{A_n}$ (equation (**21a**)) is square and that its elements are symmetric around the diagonal: therefore its transposed is identical to itself ($\boldsymbol{A_n{}^T = A_n}$). The transposed of the product of two matrices is equal to the product of the transposed matrices but in reverse order as in: $\boldsymbol{(G \cdot H)^T = H^T \cdot G^T}$ (see: http://en.wikipedia.org/wiki/Transpose). We thus also have:

$$A_n \;=\; A_n^T \;=\; (X \cdot X^T)^T \;=\; (X^T)^T \cdot X^T \;=\; X \cdot X^T \qquad (\mathbf{21c})$$

d) **Rotation of the co-ordinate system**

In the introduction of the MSA concepts above, we discussed that we aim at rotating the Cartesian co-ordinate system such that the new, rotated co-ordinate system optimally follows the directions of the largest elongation of the data cloud. Like the original orthogonal co-ordinate system of the image space, the rotated one can be seen as a collection of, say **q** orthogonal unit vectors (vector with normalized length of 1), and the columns of matrix ***U***:

$$U = \begin{pmatrix} u_{1,1} & u_{2,1} & \dots & \dots & u_{q,1} \\ u_{1,2} & u_{2,2} & \dots & \dots & u_{q,2} \\ u_{1,3} & u_{2,3} & \dots & \dots & u_{q,3} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ u_{1,p} & u_{2,p} & \dots & \dots & u_{q,p} \end{pmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \uparrow \\ \downarrow \end{matrix} \text{p} \qquad \text{(22)}$$

$$\leftarrow\!\!\rightarrow$$

q

To find the co-ordinates of the images with respect to the new rotated co-ordinate system we simply need to project the **p**-dimensional image vectors, stored as the rows in the data matrix ***X***, onto the unit vectors stored as the columns of the matrix ***U***:

$$C_{img} = X \cdot U \qquad \text{(23)}$$

Each row of the **n** x **q** matrix $\boldsymbol{C_{img}}$ again represents one full input image but now in the rotated co-ordinate system **U**. We happen to have chosen our example of **n** = 200,000 and **p** =90,000 such that **n** > **p**. That means that for the rotated co-ordinate system ***U*** we can have a maximum of **p** columns spanning the **p**-dimensional space (in other words: **q** ≤ **p**). The size of $\boldsymbol{C_{img}}$ will thus be restricted to a maximum size of **n** x **q.** We may choose to use values of **q** smaller than **p**, but when **q = p**, the (rotated) co-ordinate matrix $\boldsymbol{C_{img}}$ will contain all the information contained in the original data matrix ***X***.

e) An orthonormal co-ordinate system

In an orthonormal co-ordinate system the inner products between the unit vectors spanning the space is always zero, apart from the inner product between a unit vector and itself, which is normalised to the value 1. This orthonormality can be expressed in matrix notation as

$$U^T \cdot U = I_q \qquad \text{(24)}.$$

We have already seen the matrix ***U*** above (equation (**22**)); the matrix $\boldsymbol{U^T}$ is simply the transposed of ***U***. The matrix $\boldsymbol{I_q}$ is a diagonal matrix, meaning that it only contains non-zero element along the diagonal from the top-left to the lower right of this square matrix. All these diagonal elements have a unity value implying that the columns of ***U*** have been normalized.

f) The inverse of unit vector matrix U

The normal definition of the inverse of a variable is that the inverse times the variable yields a unity result. In matrix notation this becomes for our unit vector matrix ***U***:

$$U^{-1} \cdot U = I_q \qquad (\mathbf{25}).$$

The unit matrix $\boldsymbol{I_q}$ is again a diagonal matrix: its only non-zero elements are all 1 and are all along the diagonal from the top-left to the lower right of this square matrix. The "left-inverse" of the matrix U is here identical to its transposed version $\boldsymbol{U^T}$ (see above). We will thus use these as being identical below. We will use for example: $\boldsymbol{(U^T)^{-1} = U}$**.**

g) Conjugate Representation spaces

It may already have been assumed implicitly above, but let us emphasise one aspect of the matrix representation explicitly. Each row of the X matrix represents a full image, with all its pixel-values written in one long line. To fix our minds, we introduced a data matrix with 200,000 images (**n** = 200,000) each containing 90,000 pixel densities (**p** = 90,000). This data set can thus be seen a data cloud of **n** points in a **p**-dimensional "image space". An alternative hyperspace representation is equally valid, namely, that of a cloud of **p** points in an **n**-dimensional hyper space. The co-ordinates in this conjugate **n**-dimensional space are given by the columns of matrix ***X*** rather than its rows.

The columns of matrix ***X*** correspond to specific pixel densities throughout the stack of images. Such column-vectors can therefore be called "pixel-vectors". The first column of matrix X thus corresponds to the top-left pixel density throughout the whole stack of **n** input images. Associated with the matrix ***X*** are ***two*** hyper-spaces in which the full data set can be represented: a) the image-space in which every of the n images is represented as a point. This space has as many dimensions as there are pixels in the image; image-space is thus p-dimensional. The set of n points in this space is called the image-cloud, b) the pixel-space in which every one of the **p** pixel-vectors is represented as a single point: pixel-space is **n**-dimensional. The set of **p** points in this space is called the pixel-vector-cloud or short: pixel-cloud. (This application-specific nomenclature will obviously change depending on the type of measurements we are processing.)

Note that the pixel vectors are also the rows of the transposed data matrix $\boldsymbol{X^T}$. We could have chosen those vectors as our basic measurements entering the analysis without changing anything. As we will see below, the analyses in both conjugate spaces are fully equivalent and they can be transformed into each other through "transition formulas". There is no more information in one space than in the other! In the example we chose **n** = 200,000 and **p** = 90,000. The fact that **n** is larger than **p** means that the intrinsic dimensionally of the data matrix ***X*** here is "**p**". Had we had fewer images **n** than pixels **p** in each image, the intrinsic dimensionally or the *rank* of ***X*** would have been limited to **n**. The rank of the matrix is the maximum number of possible independent (non-zero) unit vectors needed to span either *pixel-vector* space or *image* space.

5. Mathematics of MSA data compression

The full mathematics of the PCA eigenvector eigenvalue procedures have been described in various places (for example in [Lebart 1977; 1984, Borland & van Heel 1990]). We here try to follow what we consider the best of earlier presentations with a bit of a further personal twist. We want to find a unit vector that best describes the main direction of elongation of the data cloud. "Best" here means finding a direction which best describes the *variance* of the data cloud.

a) MSA: an optimisation problem

Let direction vector "***u***" be the vector we are after (Fig **2**); the variance of image **I** that is described by a vector ***u*** is the square of the length of the projection of image **I** onto the vector ***u***, that is $\overline{OP_i}^2$. If the vector **u** is to maximize the variance it describes of the full data cloud, we need to maximize $\sum \overline{OP_i}^2$ where the sum is over all the n images in the data cloud, In doing so, we are also minimizing $\sum \overline{IP_i}^2$, the sum of the square distances of all images to the vector **u**, making this a standard least-square minimization problem.

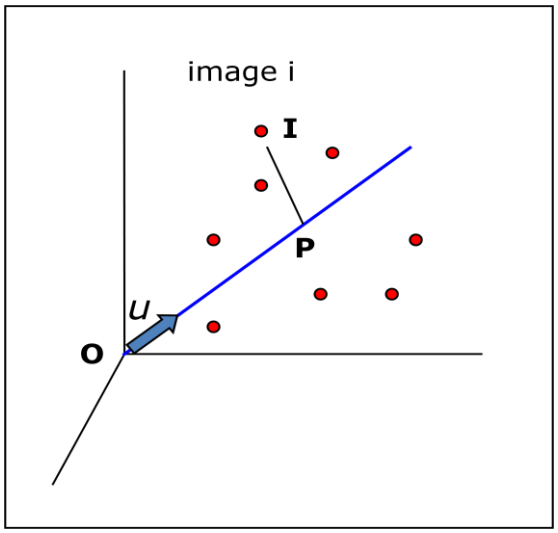

Fig 2: Finding the main variance axis of the data cloud. Each image, represented by a single point in a p-dimensional hyperspace, is projected onto the vector ***u***. Together; these points form a data "cloud". We aim at maximizing the sum of squares of these projections (inner products). The quantity we want to maximize is thus $\sum \overline{OP_i}^2$ where the sum is over all the n images in the data cloud.

We have seen above (equation (**18**)) how to calculate the inner product of the full data matrix ***X*** and a unit vector ***u***.

$$c_i = X \cdot u \qquad (\mathbf{26})$$

The sum $\sum \overline{OP_i}^2$ we want to maximize is the inner product of this resulting co-ordinate vector with itself or: $\boldsymbol{C_i^T \cdot C_i}$. We thus can write this as the variance we want to maximise:

$$\sum \overline{OP_i}^2 = \quad c_i^T \cdot c_i = \quad u^T \cdot X^T \cdot X \cdot u = \quad u^T \cdot A \cdot u \qquad (\mathbf{27})$$

Let $\boldsymbol{u_1}$ be the unit vector that maximises this variance and let us call that maximised variance $\boldsymbol{\lambda_1}$. (We will see below how this maximum is actually calculated). We then have for this variance maximizing vector:

$$u_1^T \cdot A \cdot u_1 = \lambda_1 \qquad \textbf{(28a)}$$

Since $\boldsymbol{u_1}$ is a unit vector we have (see above) the additional normalisation condition:

$$u_1^T \cdot u_1 = 1 \qquad \textbf{(28b)}$$

The data matrix has many more dimensions (keyword "*rank*") than can be covered by just its main "*eigenvector*" $\boldsymbol{u_1}$ which describes only $\boldsymbol{\lambda_1}$ of the total variance of the data set. (As mentioned above, the total variance of the data set is the sum of the diagonal elements of A, known as its *trace*). We want the second eigenvector $\boldsymbol{u_2}$ to optimally describe the variance in the data cloud that has not yet been described by the first one $\boldsymbol{u_1}$. We thus want:

$$u_2^T \cdot A \cdot u_2 = \lambda_2 \qquad \textbf{(28c)}$$

while at the same time $\boldsymbol{u_2}$ is normalized and perpendicular to the first eigenvector, thus:

$$u_2^T \cdot u_2 = 1 \qquad \textbf{(28d)}$$

and

$$u_1^T \cdot u_2 = 0 \qquad \textbf{(28e)}$$

b) Eigenvector equation in *image* space

It now becomes more appropriate to write these "eigenvector eigenvalue" equations in full matrix notation. The matrix $\boldsymbol{U}$ contains eigenvector $\boldsymbol{u_1}$ as its first column, $\boldsymbol{u_2}$ as its second column, etc. The matrix $\boldsymbol{\Lambda}$ is a diagonal matrix with as its diagonal elements the eigenvalues $\boldsymbol{\lambda_1, \lambda_2, \lambda_3}$, etc.:

$$U^T \cdot X^T \cdot X \cdot U \quad = \quad U^T \cdot A \cdot U \quad = \quad \Lambda \qquad \textbf{(29a)}$$

with the additional orthonormalisation condition:

$$U^T \cdot U = I_q \qquad \textbf{(29b)}.$$

The eigenvector eigenvalue equation (29a) is normally written as:

$$X^T \cdot X \cdot U = U \cdot \Lambda \qquad \textbf{(29c)},$$

which is the result of multiplying both sides of equation (**29a**) by $\mathbf{U = (U^T)^{-1}}$.

c) Eigenvector equation in the conjugate "pixel-vector" space

Let the eigenvectors in the space of the columns of the matrix $\boldsymbol{A}$ be called $\boldsymbol{V}$ (with $\mathbf{v_1}$ the first eigenvector of the space as its first column, $\mathbf{v_2}$ the second column of the matrix $\boldsymbol{V}$, etc.). The eigenvector equation in this "pixel-vector" space is very similar to the one above (equation (**29a**)):

$$V^T \cdot X \cdot X^T \cdot V \quad = \quad V^T \cdot A^T \cdot V \quad = \quad \Lambda \qquad \textbf{(30a)}$$

With the additional orthonormalisation condition:

$$V^T \cdot V \quad = \quad I_q \qquad \textbf{(30b)}$$

Both terms of equation (**30a)** multiplied from the left by $\mathbf{(V^T)^{-1} = V}$ again yields:

$$X \cdot X^T \cdot V \quad = \quad A^T \cdot V \quad = \quad V \cdot \Lambda \qquad \textbf{(30c)}.$$

It is obvious that the total variance described in both image space and pixel-vector space is the same since the total sum of the squares of the elements of all row of matrix $\boldsymbol{X}$ is the same as the total sum of the squares of the elements of all columns of matrix $\boldsymbol{X}$. The intimacy of both representations goes much further, as we see will below.

d) Transition formulas

Multiplying both sides of the eigenvector equation (**29c**) from the left with the data matrix $\boldsymbol{X}$ yields:

$$(X \cdot X^T) \cdot (X \cdot U) \quad = \quad (X \cdot U) \cdot \Lambda \qquad \textbf{(31)}.$$

This equation is immediately recognised as the eigenvector equation in the conjugate space of the pixel vectors, equation (**30c**) with the product matrix $(\boldsymbol{X \cdot U})$ taking the place of the eigenvector matrix $\boldsymbol{V}$. Similarly, multiplying both sides of the eigenvector equation (**30c**) from the left with the transposed data matrix $\boldsymbol{X^T}$ yields

$$(X^T \cdot X) \cdot (X^T \cdot V) \quad = \quad (X^T \cdot V) \cdot \Lambda \qquad \textbf{(32)}.$$

Again, this equation is immediately recognised as the eigenvector equation in image space (equation (**29c**)) with the product matrix $\boldsymbol{X^T V}$ taking the place of the eigenvector matrix $\boldsymbol{U}$. However, the product matrices $\boldsymbol{X \cdot U}$ and $\boldsymbol{X^T V}$ are not normalised the same way as are the eigenvector matrices $\boldsymbol{V}$ and $\boldsymbol{U}$, respectively. The matrix $\boldsymbol{U}$, is normalised through $\boldsymbol{U^T \cdot U = I}$ (equation (**29b**)), but the norm of the corresponding product matrix $\boldsymbol{X^T V}$ is given by eigenvector equation (**30a**): $\boldsymbol{(V^T X) \cdot (X^T V) = \Lambda}$. In order to equate the two we thus need to scale the product matrix by the square root of the eigenvalues:

$$U \quad = \quad X^T \cdot V \cdot \Lambda^{-1/2} \qquad \textbf{(33)},$$

and, correspondingly:

$$V \quad = \quad X \cdot U \cdot \Lambda^{-1/2} \qquad \textbf{(34)}$$

These important formulas are known as the transition formulas relating the eigenvectors in image space (***p** space*) to the eigenvectors in pixel-vector space (***n** spac*e).

e) Co-ordinate calculations

We mentioned earlier that the co-ordinates of the images in the space spanned by the unit vectors ***U*** are the product of ***X*** and ***U*** (equation (**23**)); we now expand on that, using transition formulas (**34**) and (**33**):

$$C_{img} \quad = \quad X \cdot U \quad = \quad V \cdot \Lambda^{1/2} \qquad \textbf{(35)},$$

and their pixel-vector space equivalents

$$C_{pix} \quad = \quad X^T \cdot V \quad = \quad U \cdot \Lambda^{1/2} \qquad \textbf{(36)}.$$

f) Eigen-filtering/Reconstitution formulas

We have seen that the co-ordinates of the images with respect to the eigenvectors (or any other orthogonal co-ordinate system of *p space*) are given by: $\boldsymbol{X \cdot U = C_{img}}$ (equation **(23)**). Multiplying both sides of that equation from the right by $\boldsymbol{U^T}$ (a **q** x **p** matrix) yields:

$$X^* \quad = \quad C_{img} \cdot U^T \qquad \textbf{(37a)}$$

Using equation (**34**) we can also write this equation as

$$X^* \quad = \quad V \cdot \Lambda^{1/2} \cdot U^T \qquad \textbf{(37b)}$$

This formula is known as the *eigen-filtering* or *reconstitution* formulas [Lebart 1977; 1984] as they allow us to recalculate the original data from the co-ordinate matrices and eigenvectors/eigenvalues. Note that the dimensions of ***V*** are (**n** x **q**) and those of $\boldsymbol{U^T}$ are (**q** x **p**), with **q** being maximally the size of **n** or **p** (whichever is smaller).

The reason for using a "*****" to distinguish $\boldsymbol{X^*}$ from the original ***X***, is the following: we are often only interested in the more important eigenvectors, assuming that the higher eigenvectors and eigenvalues are associated with experimental noise rather than with real information we seek to understand. Therefore we may restrict ourselves to a relatively small number of eigenvectors, or restrict ourselves to a value for **q**, of say, 50. The formulas can now be used to recreate the original data (***X***) but restricting ourselves to only that information that we consider important.

6. Mathematics of MSA with generalised metrics.

We have introduced various distances and correlation measures earlier, but in discussing the MSA approaches we have so far only considered conventional Euclidean metrics.

a) The diagonal metric matrices *N* and *M*

We have discussed above that Euclidean distances are not always the best way to compare measurements and that it may be sometimes better to normalize the measurements by their total (Chi-square distances) or by their standard deviation (Modulation distance). In matrix notation let us introduce an **n** x **n** diagonal weight matrix ***N*** that has as its diagonal elements $1/\mathbf{w_i}$, where $\mathbf{w_i}$ is, say, the average density of image $\boldsymbol{x}_i$, or the standard deviation of that image (row **i** of the data matrix ***X***). Note that then the product matrix $\boldsymbol{X' = N \cdot X}$ will have rows $\boldsymbol{x_i'}$ which will have all its elements divided by the weight $\mathbf{w_i}$. We then calculate the associated variance-covariance matrix:

$$A_p' \quad = \quad X'^T \cdot X' \quad = \quad (X^T \cdot N) \cdot (N \cdot X) \qquad (\mathbf{38}).$$

Interestingly, now all the elements of this variance co-variance matrix are normalized by the specific weights $\mathbf{w_i}$ for each original image as required for the correlations we discussed above for the Chi-square (χ^2) metrics (equation **(9)**) or the modulation metrics (equation (**13**)).

Similarly, we can introduce a diagonal (**p** x **p**) weights matrix ***M*** in the conjugate space with diagonal elements $1/\mathbf{w_j}$, where $\mathbf{w_j}$ is, the average density of pixel-vector $\mathbf{x}_j$, or the standard deviation of that pixel vector (column **j** of the data matrix **X**). Note that then the product matrix $\boldsymbol{X' = X \cdot M}$ will have columns $\boldsymbol{x'_j}$ which will have all its elements divided by the weight $\mathbf{w_j}$. Lets us now combine these weight matrices in both conjugate spaces into a single formulation. Instead of the original data matrix ***X*** we would actually like to use a normalized version ***X'*** which relates to the original data matrix ***X*** as follows:

$$X' \quad = \quad N \cdot X \cdot M \qquad (\mathbf{39}),$$

and its transposed:

$$X'^T \quad = \quad M \cdot X^T \cdot N \qquad (\mathbf{40}).$$

b) *Pre-treatment* of $\boldsymbol{X}$ with metric matrices $\boldsymbol{N}$ and $\boldsymbol{M}$

Let us now substitute these in to the PCA eigenvector eigenvalue equation:

$$X'^T \cdot X' \cdot U \quad = \quad U \cdot \Lambda \qquad (\mathbf{29c})$$

Leading to:

$$M \cdot X^T \cdot N \cdot N \cdot X \cdot M \cdot U \quad = \quad U \cdot \Lambda \qquad (\mathbf{41})$$

with the additional (unchanged) orthonormalisation normalisation constraint

$$U^T \cdot U = I_q \qquad (\mathbf{29b}).$$

With the ***N*** and ***M*** normalisations of the data matrix ***X***, nothing really changed with respect to the mathematics of the PCA calculations with Euclidean metrics discussed in the previous paragraphs. All the important formulas can be simply generated by the substitution above (equation (**39**)). For example, the co-ordinate equation (**35**) becomes:

$$C_{img} \quad = \quad X' \cdot U \quad = \quad N \cdot X \cdot M \cdot U \quad = \quad V \cdot \Lambda^{1/2} \qquad (\mathbf{42})$$

We call this pre-treatment because this multiplication of ***X*** with ***N*** and ***M*** can be performed prior to the eigenvector analysis exactly the same way as the pre-treatment band-pass filtering of the data discussed above. The procedures of the MSA analysis are not affected by pre-treatment of the data (although the results can differ substantially).

The normalisation of the data by ***N*** and ***M*** allow us to perform the eigenvector analysis from a perspective of Chi-square distances or that of modulation distances. This normalisation means that, in the 9,000,000 bicycle example for Chi-square distances, the measurements for Beijing and Cambridge fall on top of each other which is what we wanted. *However*, the fact that the weight of the measurement for Beijing is 100 times higher than that for Cambridge will be completely lost with this normalisation! That means that even for the calculation of the eigenvectors and eigenvalues of the system, the weight of Cambridge contribution remains identical to that of Beijing.

In standard (not normalised PCA), the contribution of Beijing to the total variance of the data set to the eigenvalue/eigenvector calculations would be 100^2 =10,000 times higher than that of Cambridge, thus distorting the statistics data set. (Squared correlation functions in general suffer from this problem [van Heel 1992]). It was this distortion of the correlation values that prompted the introduction of the normalisation matrices ***N*** and ***M*** in the first place. However, with the full compensation of the standard deviations of total averages through the ***N*** and ***M*** matrices we may thus have overdone what we had wanted to achieve.

c) MSA formulas with generalised metrics in "*p space*"

A more balanced approach than either the pure PCA approach or the total normalisation of the data matrix can be achieved by concentrating our efforts on a partially normalised data matrix ***X'***

$$X' \quad = \quad N^{1/2} \cdot X \cdot M^{1/2} \qquad (\mathbf{43a}),$$

and its transposed:

$$X'^{T} \quad = \quad M^{1/2} \cdot X^{T} \cdot N^{1/2} \qquad \textbf{(43b)}.$$

Substituting these in to the classical PCA eigenvector-eigenvalue equation yields:

$$X'^{T} \cdot X' \cdot U' \quad = \quad U' \cdot \Lambda \qquad \textbf{(29c)}$$

$$M^{1/2} \cdot X^{T} \cdot N^{1/2} \quad \cdot \quad N^{1/2} \cdot X \cdot M^{1/2} \quad \cdot \quad U' \quad = \quad U' \cdot \Lambda \qquad \textbf{(44)}$$

with the additional (unchanged) orthonormalisation normalisation constraint

$$U'^{T} \cdot U' \quad = \quad I_q \qquad \textbf{(29b)}$$

By then substituting $\boldsymbol{U' = M^{1/2} \cdot U}$ we obtain the eigenvector-eigenvalue equation:

$$M^{1/2} \cdot X^{T} \cdot N \cdot X \cdot M^{1/2} \cdot (M^{1/2} \cdot U) \quad = \quad (M^{1/2} \cdot U) \cdot \Lambda \qquad \textbf{(45)}$$

Which is equivalent to (multiplying left and right hand side of the equation from the left by $\boldsymbol{M^{-1/2}}$) the eigenvector-eigenvalue equation for generalised metrics [Borland & van Heel 1990; Lebart 1977; 1984]:

$$X^{T} \cdot N \cdot X \cdot M \cdot U \quad = \quad U \cdot \Lambda \qquad \textbf{(46a)}$$

However, by substituting $\boldsymbol{U' = M^{1/2} \cdot U}$, (and equivalently in the conjugate space $\mathbf{V' = N^{1/2} \cdot V}$) we deliberately choose the co-ordinate system itself to reflect the different weights of the columns and rows of the data matrix and the orthonormalisation condition now rather becomes:

$$(U^{T} \cdot M^{1/2}) \cdot (M^{1/2} \cdot U) \quad = \quad U^{T} \cdot M \cdot U \quad = \quad I_q \qquad \textbf{(46b)}.$$

d) MSA basic formulas with generalised metrics in "*n space*"

Equivalently, we obtain the eigenvector-eigenvalue equation in the conjugate space as:

$$N^{1/2} \cdot X \cdot M \cdot X^{T} \cdot N^{1/2} \cdot (N^{1/2} \cdot V) \quad = \quad (N^{1/2} \cdot V) \cdot \Lambda \qquad \textbf{(47)},$$

or, alternatively, formulated as (the result of a multiplication from the left with ($\boldsymbol{N^{-1/2}}$)

$$X \cdot M \cdot X^{T} \cdot N \cdot V \quad = \quad V \cdot \Lambda \qquad \textbf{(48a)}$$

with the associated orthonormalisation condition

$$V^{T} \cdot N \cdot V \quad = \quad (V^{T} \cdot N^{1/2}) \cdot (N^{1/2} \cdot V) \quad = \quad I_q \qquad \textbf{(48b)}.$$

e) Transition formulas with generalised metrics

For deriving the transition formulas we proceed as was done earlier for PCA derivations. Starting from the eigenvector-eigenvalue equation (45), and multiplying both sides of this equation from the left with the normalized data matrix $\boldsymbol{X'}$ $(=\boldsymbol{N^{½}\cdot X\cdot M^{½}})$ yields:

$$N^{1/2}\cdot X\cdot M\cdot X^{T}\cdot N^{1/2}\cdot(N^{1/2}\cdot X\cdot M^{1/2}\cdot M^{1/2}\cdot U) \quad = \quad (N^{1/2}\cdot X\cdot M\cdot U)\cdot\Lambda \qquad (\mathbf{49})$$

This last equation, again, is virtually identical to the eigenvector equation in the conjugate space **(**apart from its scaling):

$$N^{1/2}\cdot X\cdot M\cdot X^{T}\cdot N^{1/2}\cdot(N^{1/2}\cdot V) \quad = \quad (N^{1/2}\cdot V)\cdot\Lambda \qquad (\mathbf{48a})$$

And, again, we have a different normalisation for $\boldsymbol{N^{1/2}\cdot V}$. The latter has a *unity* norm (see equation (**48b**)), whereas $\boldsymbol{N^{1/2}\cdot X\cdot M\cdot U}$ has the norm $\boldsymbol{\Lambda}$ as becomes clear from multiplying equation (**45**) from the left with $\boldsymbol{U^{T}\cdot M^{1/2}}$ yielding:

$$(U^{T}\cdot M\cdot X^{T}\cdot N^{1/2})\cdot(N^{1/2}\cdot X\cdot M\cdot U) \quad = \quad (U^{T}\cdot M\cdot U)\cdot\Lambda \quad =\Lambda \qquad (\mathbf{50}).$$

We thus again need to normalise the "transition equation" with $\boldsymbol{\Lambda^{-1/2}}$, leading to two transition equations between both conjugate spaces:

$$V \quad = \quad X\cdot M\cdot U\cdot\Lambda^{-1/2} \qquad (\mathbf{51}),$$

and correspondingly:

$$U \quad = \quad X^{T}\cdot N\cdot V\cdot\Lambda^{-1/2} \qquad (\mathbf{52}).$$

f) Calculating co-ordinates with generalised metrics

The calculation of the image co-ordinates in *n space* as we have seen above (equation **23**):

$$C'_{img} = \quad X'\cdot U' \qquad (\mathbf{23'})$$

With the appropriate substitutions:

$$C'_{img} \quad = \quad N^{1/2}\cdot X\cdot M^{1/2}\cdot(M^{1/2}\cdot U) \qquad (\mathbf{53}),$$

and,

$$C'_{img} \quad = \quad N^{1/2}\cdot X\cdot M\cdot U \qquad (\mathbf{54}).$$

However, these co-ordinates, seen with respect to the eigenvectors $\boldsymbol{U}$, have a problem: the matrix $\boldsymbol{N^{1/2}\cdot X\cdot M}$ is only partially normalised with respect to $\boldsymbol{N}$. With the example of Beijing

versus the Cambridge bicycle density, Beijing has a hundred times higher co-ordinate values than Cambridge, while having exactly the same profile. For the generalised metric MSA we thus rather use the co-ordinates normalised fully by $\boldsymbol{N}$ and not just by $\boldsymbol{N^{1/2}}$ [Lebart 1977; 1984; Borland & van Heel 1990]:

$$C_{img} \quad = \quad N \cdot X \cdot M \cdot U \quad = \quad N \cdot V \cdot \Lambda^{1/2} \qquad \textbf{(55)}.$$

(The right hand side was derived using the transition formula $\boldsymbol{V=X\cdot M\cdot U\cdot \Lambda^{-1/2}}$ (equation (51)) multiplied from the right by $\boldsymbol{\Lambda^{1/2}}$.)

And we also have, similarly:

$$C_{pix} \quad = \quad M \cdot X^{T} \cdot N \cdot V \quad = \quad M \cdot U \cdot \Lambda^{1/2} \qquad \textbf{(56)}.$$

Using these co-ordinates for the compressed data space again puts the "Beijing measurement" smack on top of the "Cambridge measurement". How is this now different from the "total normalisation" discussed in Section 6b (above)? The difference lies in that each measurement is now not only associated with its co-ordinates with respect to the main eigenvectors of the data cloud, but each measurement now is also associated with a weight. The weight for the "Beijing measurement" here is one hundred times higher than that of the "Cambridge measurement". That weight difference is later taken into account, for example, when performing an automatic hierarchical classification of the data in the compressed eigenvector space.

7. MSA: An Iterative Eigenvector/Eigenvalue Algorithm

The algorithm we use for finding the main eigenvectors and eigenvalues of the data cloud is itself illustrative for the whole data compression operation. The IMAGIC "MSA" program, originally written by one of us (MvH) in the early 1980s, is optimised for efficiently finding the predominant eigenvectors/eigenvalues of extremely large sets of images. Here we give a simplified version of the underlying mathematics. Excluded from the mathematics presented here are the “metric” matrices N and M for didactical reasons. The basic principle of the MSA algorithm is the old and relatively simple “power” procedure (cf. [Golub & van Loan 1996]; also discussed in Wikipedia under "eigenvector power iteration"). In this traditional approach one multiplies a randomly chosen vector $\boldsymbol{r_1}$, through the symmetric variance co-variance matrix $\boldsymbol{A}$, which will yield a new vector $\boldsymbol{r_1'}$:

$$A \cdot r_1 \quad = \quad r_1' \qquad \textbf{(57)}.$$

This resulting vector is then (after normalisation) successively multiplied through the matrix $\boldsymbol{A}$ again:

$$A \cdot r_1' \quad = \quad r_1'' \qquad \textbf{(57b)},$$

and that procedure is then repeated iteratively. The resulting vector will gradually converge towards the first (largest) eigenvector $\boldsymbol{u_1}$ of the system, for which, per definition, the following equation holds:

$$\lambda_1 \cdot u_1 \quad = \quad A \cdot u_1 \qquad \textbf{(58)}.$$

Why do these iterative multiplications necessarily iterate towards the largest eigenvector of the system? The reason is that the eigenvectors "**u**" ("eigenimages") form a basis of the n-dimensional data space and that means that our random vector $\boldsymbol{r_1}$ can be expressed as a linear combination of the eigenvectors:

$$r_1 \quad = c_1 u_1 + c_2 u_2 + c_3 u_3 + ... \qquad \textbf{(59)}.$$

The iterative multiplication through the variance-covariance matrix **A** will yield for $\mathbf{r_1}$ after "k" iterations (using equation **58** repeatedly):

$$r_{1k} = c_1 \lambda_1^k u_1 + c_2 \lambda_2^k u_2 + c_3 \lambda_3^k u_3 \ + ... \qquad \textbf{(60a)},$$

or:

$$r_{1k} = \ c_1 \lambda_1^k \left(u_1 + \frac{c_2}{c_1} \left(\frac{\lambda_2}{\lambda_1} \right)^k u_2 + \frac{c_3}{c_1} \left(\frac{\lambda_3}{\lambda_1} \right)^k u_3 \ + ... \right) \qquad \textbf{(60b)}.$$

Because $\boldsymbol{\lambda_1}$ is the predominant eigenvalue, the contributions of the other terms will rapidly vanish ($|\lambda_i/\lambda_1|^k << 1$; i>1), and these iterations will thus make $\boldsymbol{r_1}$ rapidly converge towards the main eigenvector $\boldsymbol{u_1}$. The variance co-variance matrix $\boldsymbol{A}$ is normally calculated as the matrix multiplication of the data matrix $\boldsymbol{X}$ and it's transposed, $\boldsymbol{X^T}$:

$$A \quad = \quad X^T \cdot X \qquad \textbf{(61)}.$$

As was mentioned above, the data matrix $\boldsymbol{X}$ contains, as its first row, all of the pixels of image #1; its general i^{th} row contains all the pixels of image #i. The MSA algorithm operates by multiplying a set of randomly generated eigenvectors (because of the nature of the data also called "eigenimages") $\boldsymbol{r_1}$, $\boldsymbol{r_2}$, etc., through the data matrix $\boldsymbol{U}$ and its transposed $\boldsymbol{U'}$ respectively. The variance-covariance matrix $\boldsymbol{A}$ is thus never calculated explicitly since that operation is already too expensive in terms of its massive computational burden. The MSA algorithm does not use only one random starting vector for the iterations, but rather uses the full set of $\boldsymbol{q}$ eigenimages desired and multiplies that iteratively through the data matrix X, similar to what was suggested by [Clint & Jennings 1970].

In detail the MSA algorithm works as follows (**Fig 3**). The eigenvector matrix $\mathbf{U_q}$ is first filled with random numbers which are orthonormalised (normalised and made orthogonal to each other). The typical number of eigenimages used depends fully on the complexity of the problem at hand but typically is 10-100 and they are symbolised by a set of two "eigenimages" in the illustration (top of Fig **3**). Then, the inner product between these images and the **n** images of the input data set is calculated. This calculation leads to coefficient

vectors of length **n** as indicated in the left-hand side of **Fig 3**. The next step is then to calculate weighted sums over the input image stack, using the different coefficient vectors as the weights for the summing. A new set of eigenimage approximations is thus generated as shown in the lower part of **Fig 3**. New approximations are generated from this set by orthonormalisation and over-relaxation with respect to the previous set. The algorithm converges rapidly (typically within 30-50 iterations) to the most important eigenimages of the data set.

An important property of this algorithm is its efficiency for large numbers of images $\boldsymbol{n}$: its computational requirements scale proportionally to $\boldsymbol{n \cdot p}$, assuming the number of active pixels in each image to be $\boldsymbol{p}$. Many eigenvector-eigenvalue algorithms require the variance-covariance matrix as input. The calculation of the variance-covariance matrix, however, is itself a computationally expensive algorithm requiring computational resources almost proportional to $\boldsymbol{n^3}$. (This number is actually: Min ($\boldsymbol{n^2 p}$, $\boldsymbol{n p^2}$)). The MSA program produces both the eigenimages and the associated eigenpixel-vectors in the conjugate data space as described in [Borland & van Heel 1990]. One of the intuitive charms of this fast disk-based eigenvector-eigenvalue algorithm is that it literally sifts through the image information while finding the main eigenimages of the data set. The programs have been used routinely for more than 25 years, on a large number data sets consisting of up to ~1,000,000 individual images.

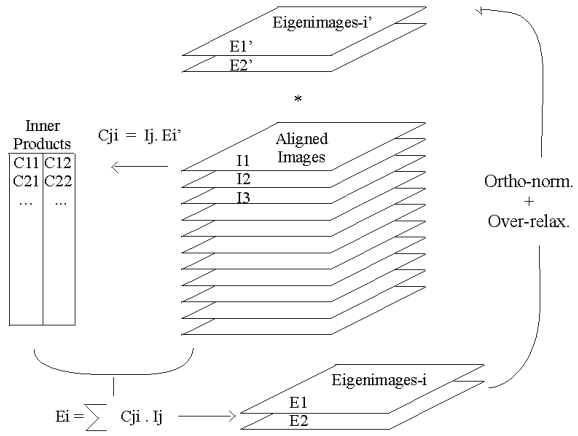

Fig 3: The working of the iterative MSA eigenvector-eigenvalue algorithm. In essence the algorithm first fills a matrix $\boldsymbol{U}$ with orthonormalised random column number vectors (E_1', E_2', etc.). This matrix is then multiplied through the data matrix $\boldsymbol{X}$ containing the set of images I_1-I_n as rows. This multiplication leads to a matrix of coefficients $\boldsymbol{C}$ which is then multiplied through the transposed data matrix $\boldsymbol{X^T}$ leading to (after normalization) a better approximation of the eigenvector matrix. In essence the algorithm performs a continuous set of iterations through the eigenvector equation (**29a**) $\boldsymbol{U'' = \Lambda^{-1} X^T . X . U'}$. (For the equivalent including the influence of generalized metrics, we have from equation (**46a**): $\boldsymbol{U'' = \Lambda^{-1} . X^T . N . X . M . U'}$)

8. Parallelisation of the MSA algorithm.

In spite of its high efficiency and perfect scaling with the continuously expanding sizes of the data sets, compared to most if not all conventional eigenvector-eigenvalue algorithms, the single-CPU version of the had become a serious bottleneck for the processing of large cryo-EM data sets. The parallelisation of the MSA algorithm had thus moved to the top of our priority list. We have considered various parallelisation schemes including the one depicted in **Fig 4**; this mapping of the computational problem onto a cluster of computers was indeed found to be efficient.

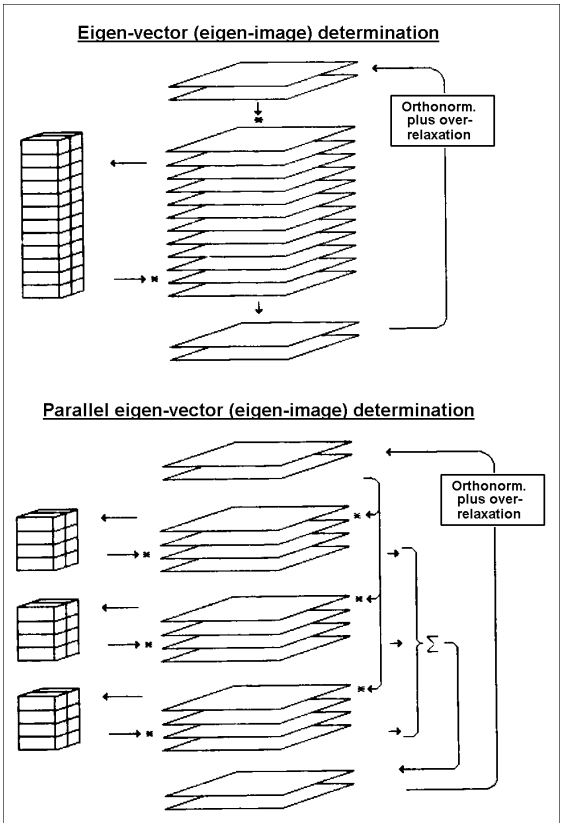

Fig 4: **Parallelisation of the MSA algorithm**. Note that for an efficient operation it is essential that the part of the input data matrix ***X*** that is of relevance to one particular node of the cluster, is indeed always available on a high-speed local disk.

As was discussed before, equation (**46a**) (see also the legend of **Fig 3**) is the full matrix equivalent of:

$$u_j' = X^T \cdot N \cdot X \cdot M \cdot u_j \quad ; \qquad u_j' = u_j \cdot \lambda_j \qquad \textbf{(62)},$$

In which, $\boldsymbol{u_j}$ is one single eigenvector and $\boldsymbol{\lambda_j}$ the corresponding eigenvalue. Having stated this, a direct approach for parallelisation of the calculation of equation (**46a**) could be the exploiting of the independence of calculations of all $\boldsymbol{u_j}$ since each eigenvector $\boldsymbol{u'_j}$ depends only on one eigenvector $\boldsymbol{u_j}$. Although it is a clear possibility and is quite straightforward to implement, this scheme does not represent an effective approach. Since there may typically be more available processors than the number of eigenvectors needed for representing the dataset, this approach would lead to a waste of processing time with many processors running idle. Moreover, all processors would need access to the full image matrix ***X***. This approach is thus likely to create severe I/O bottlenecks.

To analyse another possible parallelisation schemes, let us write equation (**46a**) as:

$$u'_{ij} = \sum_{s=1}^{n} x_{si} n_s \sum_{t=1}^{p} x_{st} m_t u_{tj} \qquad \textbf{(63)},$$

where $\boldsymbol{u_{ij}}$ is element **i** of eigenvector $\boldsymbol{u_j}$, $\mathbf{x_{ij}}$ is pixel ***j*** of image **I**, and $\mathbf{n_s}$ and $\mathbf{m_t}$ are the diagonal elements associated to ***N*** and ***M*** metrics, respectively. This equation can be expanded into the form:

$$u'_{ij} = \sum_{s=1}^{k} x_{si} n_s \sum_{t=1}^{p} x_{st} m_t u_{tj} + \sum_{s=k+1}^{2k} x_{si} n_s \sum_{t=1}^{p} x_{st} m_t u_{tj} + \sum_{s=2k+1}^{3k} x_{si} n_s \sum_{t=1}^{p} x_{st} m_t u_{tj} + \ldots + \sum_{s=n-k+1}^{n} x_{si} n_s \sum_{t=1}^{p} x_{st} m_t u_{tj} \quad \textbf{(64).}$$

The equation above suggests a possible parallelisation scheme over the image dataset instead of the eigenvectors (**Fig 4**). Each processor calculates a partial solution for $\mathbf{u'_{ij}}$ based on a subset of **k** images. After the complete calculation of $\boldsymbol{U'}$, the further steps of the algorithm are computed in a non-parallel way. The algorithm thus consists of a parallel and a non-parallel part.

The implementation of the algorithm can be structured in a master/slave architecture, where the master is responsible for all non-parallel tasks and for summing the partial results of each node. During each iteration of the algorithm, each node needs to read data from the image matrix $\boldsymbol{X}$ thus potentially creating a network bottleneck. However, since each node in this parallelisation scheme needs only to access a part of the $\boldsymbol{X}$ matrix, distributing the partial datasets of input data over the local disks efficiently, parallelises the I/O operations. A fine tuning is in progress to further minimise the hitherto not parallel part of our parallel MSA algorithm. Suppose that in a single CPU environment the non-parallelised code consumes 10% of the run time of the overall MSA calculation and the parallelisable code 90%, In a 100-CPU environment (assuming 100% efficiency) the 90% would drop to 0.9% where as the rest would remain at 10% (of the single CPU timing). This implies that from the user perception, the parallel code accounts for 10% of use of the full cluster and 90% is "wasted" on the non-parallel code. Fine tuning the program to minimise the ratio of the non-parallel code versus the parallel code, is thus time well invested in term of minimising the run time (and thus of the "user satisfaction") of algorithms such as the "parallel MSA".

9. Exploiting the compressed information in Factor Space

What we wish to achieve with the MSA approaches is to better organise the intrinsic information of large noisy data sets and thus make them more accessible. The determination of the eigenvectors (eigenimages) by itself does not change or reduce the total image information in any way. It is merely a rotation of the co-ordinate system in a special direction, such that the first eigenvector covers most of the variance in the data cloud, the second, orthogonal to the first, covers most of the remaining variance of the data cloud, not covered by the first eigenvector, etc. If one would find *all* the eigenvectors of the data cloud (all the eigenvectors of the variance-covariance matrix; all **p** or **n** of them, whichever is the smaller number) there would be no loss of information at all. Of course, even without any loss of information the *most important* information is associated with the first eigenvectors/eigenimages that describe the strongest variations (in terms of variance) within the data set.

When one decides to consider only the first so-many eigenvectors and thus decides to ignore all higher eigenvectors of the system that is where political decision is made between what is signal and what is noise. We actually decide here that the percentage of the overall variance of the data set (the "trace") covered by these so-many main eigenvectors, suffices.

Once this decision is made of what is necessary and enough to one tackle the problem we have achieved a sometimes spectacular level of data compression. The problem remains of how to best exploit that information in the compressed data space. Having concentrated on the mathematics and algorithmic aspects of the eigenvector-eigenvalue aspect of the MSA approach, we here will just give a relatively short account of this most fascinating aspect of the methodology. The first EM data sets that were ever subjected to the MSA approach were simple negative-stain contrasted specimens where the differences among the images were directly visible in the micrographs. Indeed the data sets were chosen carefully for being simple in order for the problem to be solvable with the limited computing resources available in those days.

10. Visual classification in a low-dimensional factor space

The first data set that was ever analysed by the MSA approach was an artificial mixture dataset from two different species of hemoglobins. The first analyses were performed with the "AFC" program developed by Jean Pierre Bretaudière from a program originally written by JP Benzécri (see Appendix for details). The 16 molecular images of the giant annelid hemoglobin of *Oenone fulgida* visibly had extra density in the centre compared to the 16 molecular images of well known giant hemoglobin of the common earthworm *Lumbricus terrestris* which rather had a “hole” in the centre of its double-ring geometry (see appendix). The separation of the two species, artificially mixed for this analysis, was so obvious that one immediately moved on to a single data set with real internal variations (the example below). This mixed hemoglobin data set, however, is still being used for illustration purposes (Fig 4.4 of [Frank 2006]).

The first real data set with genuine variations among the molecular images was a preparation *Limulus polyphemus* hemocyanin half-molecules, each consisting of four hexamers produced by controlled dissociation of the full molecule. The data sets were by today's standards extremely small with only 46 images in this case. Each 64x64 image was then low-pass filtered and coarsened to a mere 32x32 pixel format. Nevertheless the fact that the image information could be compressed logically to points on a two-dimensional plane (**Fig 5**) was a tremendous breakthrough. The logical grouping of those points on this factorial "map" immediately suggested which images to average together in order to suppress the noise present in the individual images. Which images were grouped together, however, was a direct human decision, in spite of the proud first line of the abstract stating: "We have developed a new technique of analysis that allows *automatic classification* of molecule images according to subtle differences."

In those early days in computing, the address space of the computers was minimal by today's standard. The 16-bit PDP-11/45 computer on which this work was performed would only address 2^{16} bytes = 64 Kbytes of memory. It was thus difficult to fit any serious computational problem in the central memory of that generation of computers! The critical memory requirement for the eigenvector-eigenvalue calculations was the “core” needed for the square variance-covariance matrix which, practically limiting the number of "particles" to around 50 at most. In the IMAGIC implementation of Bretaudière's AFC program (see appendix) the limit was slightly over 100 molecular images (see the *Limulus polyphemus* hemocyanin double-hexamer analysis in [van Heel & Keegstra 1981]). That larger number of

particles was purely due to the fact that the NORSK DATA Nord-10 computer addressed 16-bit word rather than the 8-bit words of the PDP-11/45, a doubling of the usable memory. With the computer hardware limiting the complexity of problems that could be reasonably studied, the limitations of being able to only look at problems that were intrinsically two-dimensional, was not immediately felt as a limitation. It was more frustrating that the averaging was over classes that still were a subjective choice of the user after the elegant data compression.

A new generation of 32-bit computers – with a much larger address space – came on the market (such as the Digital Equipment VAX-780) allowing one to handle a much larger data sets with a significantly higher level of complexity. There were some efforts to map more complicated cases nonlinearly onto just a 2D plane in order to visualise the underlying structure. Although such approaches may relieve the problem somewhat, what was really needed was a more abstract approach in which all information in the compressed data space is structured automatically. The development of automatic classification procedures operating on a, say, 60-dimensional compressed data space became an absolute necessity**.**

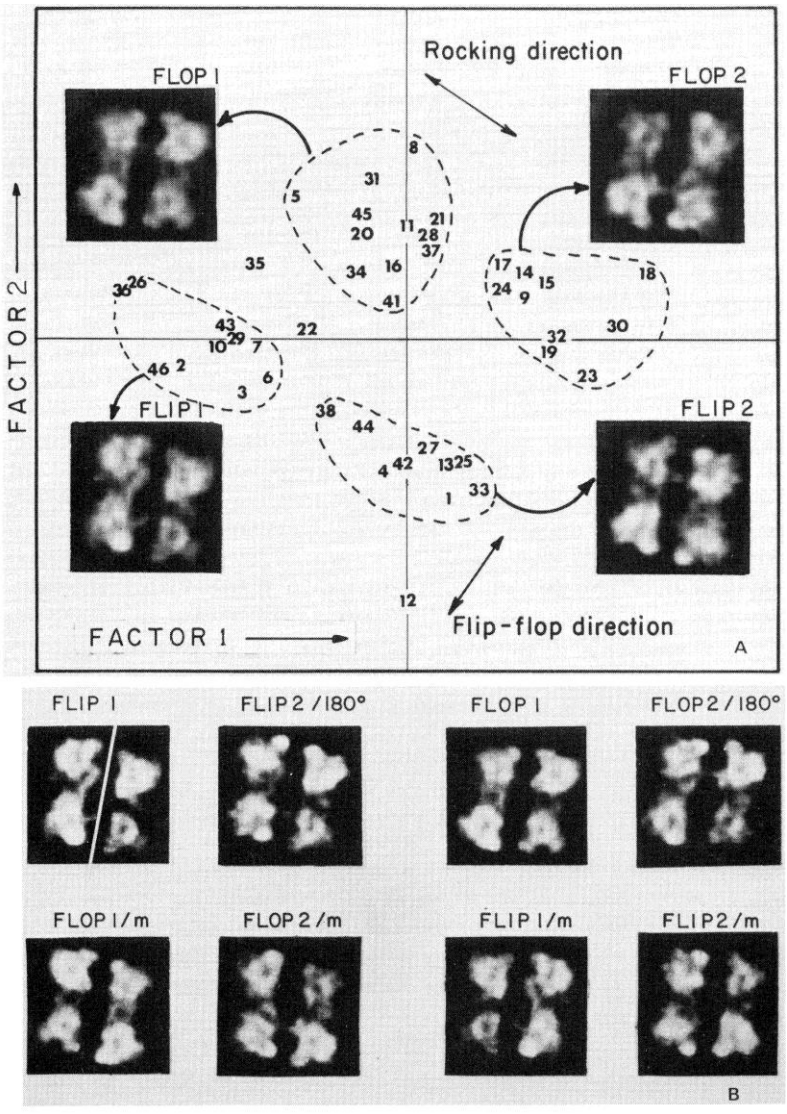

Fig 5. Visual Classification:
Historical correspondence analysis of negatively stained "4x6" half-molecules of the hemocyanin of *Limulus polyphemus.* The two two-hexamer rods constituting this molecule are shifted with respect to each other thus forming a parallelopid which looks different in face-up and face-down position ("flip" and "flop").
The four hexamers constituting the 4x6-mer were, however, apparently not in a single plane which leads to two stable "rocking" positions where 3 out of 4 hexamers are in touch with the carbon support film, and one not and is therefore stained less. This effect splits both the "flip" and the "flop" groups in two "rocking" positions.
A serious complication in interpreting this two-dimensional factor map was that alignment schemes used, introduced a 180º ambiguity in the alignment. ([van Heel & Frank 1981])

11. MSA Automatic classification algorithms

To eliminate the subjectivity of visual classification (which is simply impossible when the number of factors to be taken into account is larger than two or three), automatic classification schemes were introduce in EM. These schemes emerged in the general statistical literature in the 1960s and 1970s, and were first used in electron microscopy in the early 1980s. There are many algorithms available each with its advantages and disadvantages. The three listed below have been or are important in single particle cryo-EM

a) "K-means" algorithms

In the first class of algorithms, one chooses a number of "***k***" classes into which one wishes to use to subdivide the full data set (of ***n*** images) and then selects at random ***k*** from the ***n*** elements to serve as the first classification seeds. Then one loops over all ***n*** elements of the data set and assign each element to one of the ***k*** seeds based on a closest proximity. After one iteration, each of the k original seeds shift to the centre of mass ***k′*** of all those elements of the total of ***n*** that were found to be closest to the original random seed ***k***. All original ***n*** elements of the data set are then classified against the ***k′*** new classification centres and the process is the repeated. This type of algorithm is now known as a "**k-means**" algorithm [Bock 2008]. Its first use in EM was in [van Heel & Stöffler-Meilicke 1982] using Diday's "dynamic clouds" approach [Diday 1971; van Heel 1989; Frank 2006]. In Diday's approach [Diday 1971] the pure k-means algorithm is run multiple times with different random seed starting points, in order to find elements that always group together and thus form stable clusters. However, in spite of the k-means approaches being fast and useful, their use was found somewhat disappointing because the results depend strongly on the random seeds chosen. There is also an inherent contradiction between the large number of seeds needed to sample the information space sufficiently fine to not miss relevant information hidden in a small corner of factor space, and the low number of classes desired to allow us humans to better understand the results. For a more extensive discussion of the disadvantages and advantages of k-means algorithms see references [van Heel 1989; Bock 2008].

b) Hierarchical Ascendant Classification (HAC)

The HAC technique is a "from-the-bottom-up" classification approach in which each element of the data set is first considered as a "class" by itself. Each of these individual starting classes can be associated with a mass depending on the choice of metric. Individual classes are then merged in pairs, based on a merging criterion, until finally one single large class emerges, containing all the elements of the original data set. The history of the classification procedure (that is, which classes merge together at which value of the aggregation criterion) is stored in a classification or merging "tree". The user chooses the desired number of classes at which the tree is to be cut. As a merging criterion, the minimum added intra-class variance or Ward criterion is normally used [Ward 1982; Benzécri 1980; Lebart 1984]. The algorithm is aimed at minimising the internal variance of the classes (“intra-class variance”) while at the same time maximising the inter-class variance between the centres of mass of the class averages. At any level of the procedure, two classes are merged to form a new, larger class if the increase in total intra-class variance associated with their merging is the lowest possible at that level.

The *Added-Intra-class-Variance* associated with the merging of class i with class j is:

$$AIV_{ij} = \frac{(w_i \cdot w_j)^2}{(w_i + w_j)} \cdot d_{ij}^{\,2} \qquad \textbf{(65).}$$

In this formula $\mathbf{d_{ij}^2}$ is the (now) Euclidean square distance between the classes **i** and **j** having masses (weights) $\mathbf{w_i}$ and $\mathbf{w_j}$ respectively (Appendix 2 of [van Heel 1984a]). To obtain a predefined number of classes from the process, one then cuts the merging "tree" at the appropriate level. The advantage of this type of algorithm is that it provides a logical and consistent procedure to subdivide the data set into various numbers of classes, whatever number makes sense in terms of the specific problem at hand. The algorithm helps visualising the inherent structure of the data by showing – at all levels – which classes are to merge with which classes. The first use of HAC schemes in electron microscopy was in [van Heel 1984a, b]. Because of the explicit influence of the masses in HAC (equation (**65**)) the approach is well suited for the generalised metrics including the Chi-square metric and the modulation metric.

A disadvantage of the HAC algorithm [van Heel 1984a,b; 1989; Borland & van Heel 1990] compared to the K-means schemes, however, are the computational efforts required. For smaller data sets of up to ~10,000 images/elements, the computational requirements are negligible compared to those of the eigenvector-eigenvalue calculations themselves. However, since the computational requirements of the algorithm grow proportionally to ~ $\mathbf{n^2}$, (with **n** the number of images in data set) for larger data sets of, say, more than 200,000 elements, the computational requirements become excessive and often exceed the requirements for the eigenvector calculations.

Although, at every merging step, two classes are merged that lead to a minimum AIV contribution, this fact is also a fundamental limitation: if two elements are merged into one class at an early stage of the procedure, the elements will always remain together throughout all further HAC classification levels, whereas if their marital ties were weaker, a lower intra-class variance minimum could easily be obtained [van Heel 1989]. Also, the merging of two classes at a later level of the HAC algorithm may locally cause a large increase in intra-class variance. A simple yet most effective post-processor was designed to deal with the problem that any HAC partition is typically far from a local minimum of the total intra-class variance.

c) Moving elements refinement

The moving-elements approach is a post-processor to refine and consolidate an existing partition. Starting point is typically the partition obtained with HAC based on the Ward criterion discussed above. The partition is refined, to reach a "deeper" local minimum of intra-class variance, by allowing each member of each class to migrate to any other class where it is happier in terms of the very same added intra-class variance criterion [van Heel 1989; van Heel & Stöffler-Meilicke 1985]. Using the HAC partition as a starting point, each element is extracted from its HAC class and the AIV distances relative to all other classes are calculated for that particular element. If the minimum of these AIV values is lower than the element's AIV distance to its current class, it is extracted from its current class and placed into the other one. The statistics of both its old and its new class are updated and the algorithm proceeds with the next element.

After one cycle, many of the classes have changed; hence the procedure is iterated until no further moving of elements is observed. The partition thus obtained is a real (and significantly deeper) local minimum of the total intra-class variance than that obtained directly by the HAC in the sense that no single element can change class-membership without increasing the total intra-class variance of the partition. We call this algorithm "moving elements consolidation" (or refinement) as opposed to the "moving centres consolidation" proposed independently by [Morineau & Lebart, 1986], which is a pure K-means post-processor to the HAC partition based on the same basic idea that HAC partitions are prone to improvement. The HAC scheme in combination with the moving elements post-processor has emerged as in general the most robust classification scheme. Further possible classification schemes/refinement schemes have been discussed in [van Heel 1989]. Some other classification schemes have been reviewed in [Frank 2006].

After the classification phase, all noisy molecular images that have been assigned to the same class in the classification phase are averaged together. This averaging of images leads to a large improvement in SNR (Signal-to-Noise-Ratio). The SNR improves proportionally to the number of images averaged, under the idealised assumption that all images averaged are identical apart from the noise, which is different from image to image. The new class averages may be used as references for a new alignment and classification iteration round. After a few iterations, good class averages with improved signal-to-noise ratios can be obtained. The high SNR values obtained are of great importance for an accurate assignment of Euler angles to these projection images by techniques such as "projection matching" [van Heel 1984; Frank 2006] or by "angular reconstitution" [van Heel 1987].

12. MSA symmetry analysis

Another example of the use of MSA is the unbiased analysis of the main symmetry properties of a macromolecular complex. Earlier symmetry analysis approaches were based on finding the predominant rotational symmetry components of one single image at a time [Crowther & Amos 1971]. That single image could then be used to align all particles of the data set. However, when aligning a whole set of images with respect to one single image with a strong, say, 6-fold symmetry component, the 6-fold symmetry property will be implicitly imposed upon the whole data set by the reference-bias effect [Boekema 1986; Stewart & Grigorieff 2004] during the alignment procedure. When an average is then obtained from the aligned images, the 6-fold symmetry component becomes *overwhelmingly* present, but that serves only as a demonstration of the reference bias effect, and certainly not as a proof of the real symmetry of the complex being studied. Numerous papers have appeared in the literature in which this “self fulfilling prophecy” approach was used for determining the “true” symmetry of an oligomeric biological structure.

A methodologically clean approach for determining the strongest symmetry component in a data was proposed, which entirely avoids the symmetry bias resulting from any explicit or implicit rotational alignment of the molecular images [Dube 1993]. In this approach, only a translational alignment relative to a rotationally symmetric “blob” is performed for centring particles. The rotational orientation of all molecules remains arbitrary, and will thus often be the main source of variation among the images of the data set. MSA eigenvector analysis will

then find those main eigenimages of the system and these reflect the symmetry properties of data elegantly.

The particles used here (**Fig 6**) resulted from an automatic particle selection over a stack of micrographs of the hemoglobin of the common earth worm *Lumbricus terrestris.* We use these data as a high-contrast general testing standard [van Heel 2000]. The particle picking program was used in cross-correlation mode with a rotationally-symmetric reference/template image (see insert in **Fig 6**). This particle-particle picking procedure yields centred particles equivalent to the translational alignment with a rotationally symmetric average used in the original paper [Dube 1993].

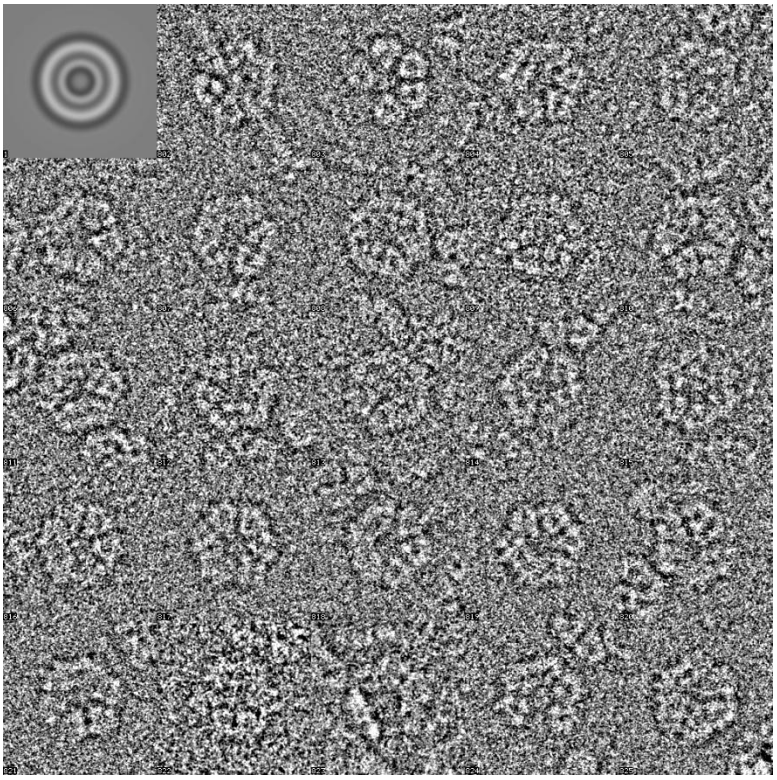

Fig 6. Some of the ~7,300 particles selected automatically from a set of micrographs using the automatic particle picking program PICK_M_ALL (IMAGIC). The rotationally symmetric template image shown in the top-left corner of this illustration was used to find the particles by cross correlation. This procedure yields particles that are centred with respect to the reference image. These particles have no bias to any reference image in a rotational direction and such a data set is ideal input for subsequent MSA-based symmetry analysis (Fig **7**)

The total of some 7,300 particles were submitted to MSA eigenvector analysis. The first eigenvector in this analysis (**Fig 7**) is almost identical to the reference image used to pick the particles. The two next eigenimages are essentially two identical 6-fold symmetrical images rotated by a small angle with respect to each other. A six-fold symmetrical structure repeats itself after a rotation over 60º. These two eigenimages 2 and 3, are rotated by exactly 15º degrees with respect to each other. The eigenimages 2 and 3 consist of sine waves along the tangential direction of the images and along each circle around the centre of the eigenimage we will see 6 full periods of that sine wave. One period of the sine wave corresponds to a 60º arc on the circle. The 15º rotation between the two eigenimages mean that the sine waves at each radius of the two images are $\pi/2$ out of phase with each other, like a sine and a cosine function. Note also (from the eigenvalue plot in **Fig 7**) that they have an associated eigenvalue of more than twice the magnitude of all subsequent eigenvalues.

Interestingly, the next two eigenimages (4 and 5) show a 12-fold symmetry (the first higher harmonic of the main six-fold symmetry component of the data set (eigenimages 2-3). Again, the two eigenimages are rotated by $\pi/2$ of one tangential sine period; in other words, the two images are rotated by 7.5º with respect to each other. Eigenimages 6 and 7 exhibit 18-fold

symmetry (rotated by 3.75º) and represent the 3rd harmonic of the 6-fold symmetry component.
This MSA-based methodology for determining the strongest symmetry component in a data entirely avoids the symmetry bias resulting from any explicit or implicit rotational alignment of the molecular images. The translational alignment with respect to a rotationally symmetric reference image does not introduce any bias in the data with respect to symmetry. The translational alignment, that was performed explicitly in [Dube 1993], here is implicit in the automatic particle selection procedure with respect to a rotationally symmetric reference image.

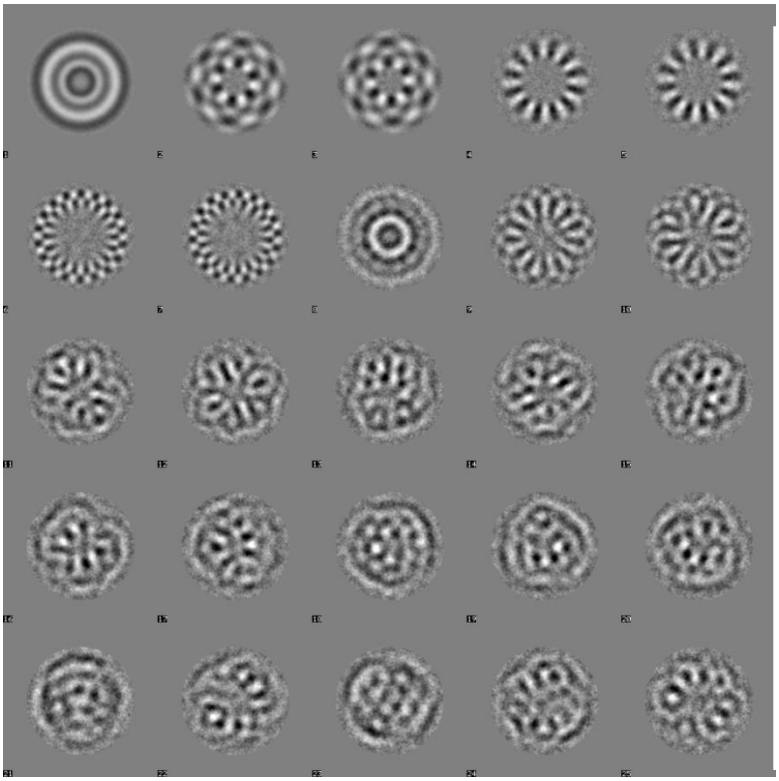

Fig 7. The first 25 Eigenimages of the data set of ~7,300 particles (**Fig 6**) together with a plot of the associated 10 first eigenvalues. Eigenimages 2 and 3 exhibit perfect 6-fold symmetry, and are virtually identical to each other apart from a small rotation. They are rotated by exactly 15º degrees with respect to each other meaning that the tangential sine waves at each radius of the two images are π/2 (90º) out of phase with each other. Note from the eigenvalue plot, that the first eigenvector covers ~2% of the total variance of the data set, and that the first 10 eigenvectors together cover around 4.5% of that total variance

```
Eigenvalues (measured in percentage of total variance)

    1         2.012 |************************************************************
    2         0.569 |*****************
    3         0.543 |****************
    4         0.253 |*******
    5         0.250 |*******
    6         0.208 |******
    7         0.206 |******
    8         0.165 |*****
    9         0.148 |****
   10         0.144 |****
```

13. Alignments and MSA

Alignments of the images with the data matrix ***X*** change the MSA analysis in fundamental ways. Interestingly however, alignments do not change the overall variance of the data set. A rotation of an image merely shifts around the pixel densities within a row of the data matrix ***X***; the total variance of the measurement, however, does not change. (This is exactly true as long as no non-zero pixels are rotated or shifted out of the part of the image that is active

during the MSA analysis). The variance of each image is the corresponding diagonal element of the variance-covariance matrix ***A*** and hence these diagonal elements remain unchanged.

The total variance of the data set (the *trace* of ***A***) is the sum of the diagonal elements of matrix ***A*** (equation (**21a**)). What does change during the alignment procedures is that certain images within the data matrix will become better aligned to each other and will sense a higher co-variance among them (the off-diagonal elements of matrix "***A***").

We will here apply an interesting and illustrative form of alignment, namely: "Alignment by MSA", (called "alignment by classification" in [Dube 1993]) to the worm hemoglobin data set introduced above (**Fig 6**). Having noticed that eigenvectors 2-7, all are associated with 6-fold symmetry or the harmonics thereof, we choose to align the (already centred) data set - in a rotational sense only - with respect to eigenimage #2 (**Fig 8**). The rotation alignment was restricted in range of -30º to +30º in order to leave the 6-fold-symmetry character of the data set undisturbed. Eigenimage #2 is pasted in the top-left corner of the illustration, whereas all the other images in this figure are the rotationally aligned versions of the centred particle of the original data set (**Fig 6**).

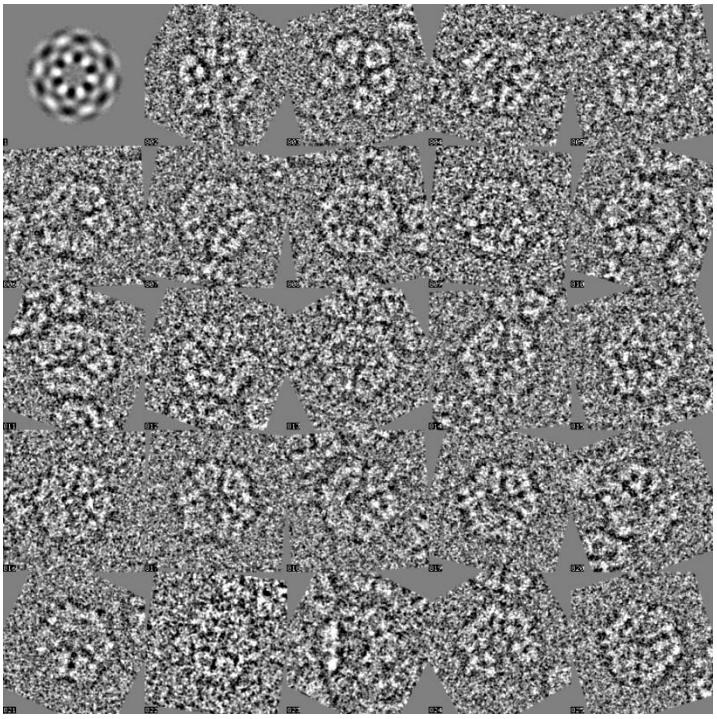

Fig 8: Some of the ~7,300 automatically selected particles introduced above (**Fig 6**), but now rotated using the second eigenimage as a reference for a rotation-only alignment. The rotation was restricted to -30º to +30º to stay within the 6-fold symmetry properties of the data set. The now rotationally aligned data set was again submitted to MSA eigenvector analysis; the results of that analysis are shown in **Fig 9**.

We have here aligned the full (centred) data set rotationally with respect to one of the two main eigenimages associated with the 6-fold symmetry property of this hemoglobin. In doing so, we have concentrated a much larger percentage of the total variance of the data set into the first few eigenvectors of the system as is seen from the total sum of the first 10 eigenvalues before (4.3%; **Fig 7**) and after the rotation-only alignment (6.6%; **Fig 9**). As was mentioned above, an alignment does not change the variance of each individual image and thus does not change the total variance of the data set.

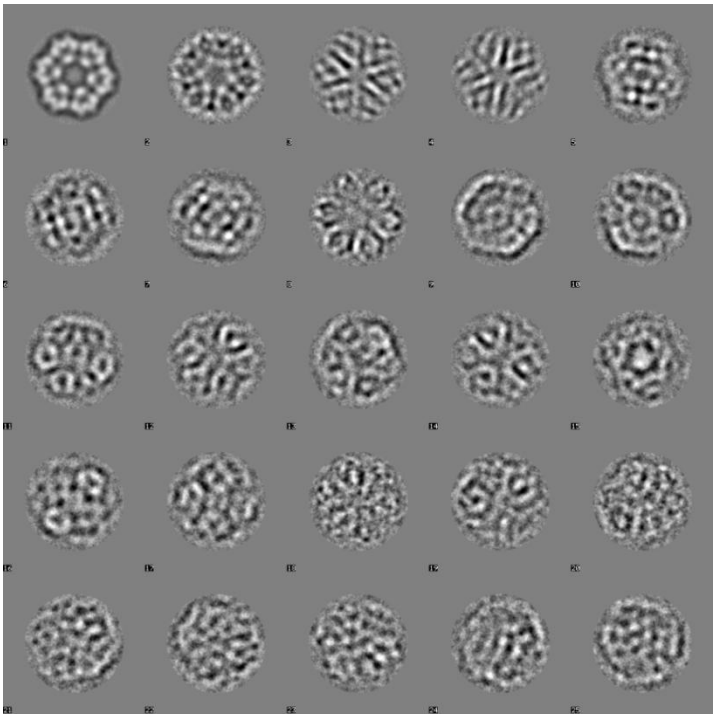

Fig 9: MSA Eigenvector analysis of the centred data set, but now aligned rotationally with respect to the second eigenimage of the first MSA analysis (**Fig 7**). The first eigenvector now is clerly recognisable as a top view of the *Lumbricus terrestris* hemoglobin. It alone now covers ~4.3% of the total variance of the data set (compared to the 2.01% the first eigenvector covered in the first analysis (**Fig 7**). The first 10 eigenvectors together cover around 6.6% of the total variance of the data set.

```
Eigenvalues (measured in percentage of total variance):

    1         4.335 |************************************************************
    2         0.396 |*****
    3         0.367 |*****
    4         0.346 |*****
    5         0.209 |***
    6         0.209 |***
    7         0.203 |***
    8         0.180 |**
    9         0.173 |**
   10         0.172 |**
```

Concentrating the information (variance) into the first few eigenimages means that more eigenvectors will become statistically significant above the background random noise level. With the relatively small data sets from the past, alignments were an absolute necessity for obtaining statistically significant results. Today, however, and especially with the powerful new parallel MSA algorithms, we can afford to largely increase the overall size of the data set. The rationale for using larger data sets is simple: using ten times more images in a data set means using ten times more information! In doing so, we can achieve statistical significance without the risk prejudicing the data sets by introducing reference bias [Boekema 1986; Stewart & Grigorieff 2004]. This increased level of objectivity is especially important when trying to find objectively find different conformations in a heterogeneous population of molecular images.

14. MSA and heterogeneous data sets

Mixed populations of molecular images were originally the curse of single-particle electron microscopy. Originally, one would try to work with an as homogeneous as possible population of macromolecules. The basic assumption for single particle cryo-EM was that all molecules were the same, apart from the fact that each molecule could exhibit a different orientation and be in a different position in the sample. When the data were heterogeneous or

flexible, however, (and that heterogeneity is not recognised) that results in a deterioration of the quality of the resulting 3D structure since one is then averaging "cows" and "horses". When, on the other hand, one knows how to separate the molecular images into structurally homogeneous subgroups, a whole new world of possibilities opens. One can then study a mixture of closely related structures and from that commence to understand the biological reasons for the distinct 3D structures found in a sample. The name "4D" cryo-EM has been used in this context.

The first example of the challenging possibilities came from a cryo-EM sample of the *E. coli* 70s ribosome, in complex with Release Factor 3 [Klaholz 2004]. It was found the structure simply did not converge towards a high-resolution structure no matter how much effort was invested. However, once the data set was allowed to refine (by multi-reference alignment) towards different 3D reconstructions simultaneously, the sample turned out to consist of a mixture of ribosomes in two different conformational states. The ribosomal complexes were found to co-exist in the pre-translocation and post-translocational states of the 70S ribosome. Thet"RF3" was also found in two different conformations depending on the conformation of the 70S ribosome itself. Once the data set had been separated into conformational subsets, by local MSA of specific projections, each subset of images led to a separate 3D structure exhibiting a much higher resolution than the resolution on the overall 3D reconstruction. A further example, where size variations among the particles was found to be a limiting factor, is given by [White 2004].

Another possibility to study 3D structural variation is to generate a large number of 3D reconstructions and to study those by means of MSA the way 2D molecular images were studied above. This has always been possible with the MSA programs for those who were sufficiently intimate with the software, by re-defining the size of the input images to actually cover a full input 3D. More recently, the IMAGIC system has been re-organised to manage 3D volumes with the same ease and transparency as the system handles sets of 2D images (publication in preparation). This has greatly facilitated handling these huge intricate datasets.

A quality criterion for the 4D MSA types of analysis is the resolution achieved in the subset 3D reconstructions. As usual the resolution is quantified by Fourier Shell Correlation ("FSC", [Harauz 1986; van Heel 2005]). There are two conflicting issues here. It is generally true that larger datasets lead to higher resolution results than smaller datasets. An increase of resolution, while decreasing the number of raw images entering in each specific 3D reconstruction, is an important indication for the population of images indeed to belong to different structural subsets. The separation of large data sets into structurally homogeneous subsets must be seen as a major challenge in current biology and this area is in constant flow (see below). We do not want to go into much detail here but rather point the reader to a recent general review [Leschziner & Nogales, 2007].

15. Discussion

The main goal of the biological sciences is to make sense of the massive and very noisy "data" in terms of elucidating the underlying general principles. No better example than the data sets collected in cryo-EM which are probably among the largest and noisiest currently being collected in the biomedical sciences. Over the past three decades, multivariate statistical analysis approaches have been very successful in helping us sort out complex EM

data sets in many different ways, and a few examples have been discussed in this paper. However, we have not yet seen the full potential of the MSA approaches exploited due the costs of the computing effort. This high cost has hampered the use and the development of potentially very useful further MSA approaches such as the "alignment by MSA" discussed above.

How can that be that the computational effort was simply not affordable? After all, one of the most dramatic developments in science over the past three decades since the first use of the MSA approach in EM, has been the increase in speed in general computing. In spite of massive computing resources becoming available prolifically, we have seen an even more rapid increase in the size and complexity of the EM data sets. The size of the data sets themselves have been limited not by what one would wish to analyse, but rather by what was practical to analyse. The high-resolution scanning of (analogue) micrographs into digital form was a tedious activity taking up to hours of labour for a single micrograph. With robotic scanning devices for film now becoming available, or with automatic 24/7 data collection directly on the microscope – using on-line 4k x 4k (or even 8k x 8k) CCD/CMOS cameras – the data-collection bottleneck has been alleviated. Therefore, more than ever, the speed of the MSA computations has become a very serious bottleneck limiting its use in cryo-EM. The recent parallelisation of MSA programs (operating in the IMAGIC software system [van Heel 1996; van Heel 2000]) now opens a whole range of new possibilities by speeding up the process by orders of magnitude, exploiting the power of hundreds of CPUs simultaneously.

There are other reasons why the MSA have not always received the appropriate attention and an important reason is ignorance. The MSA approaches are not the simplest of techniques to understand and to get acquainted to for the potential biological user. The learning curve is steep and users tend to avoid investing the time necessary to understand the approach. More serious, however, is that many software systems in use in cryo-EM do not even have MSA options, and those that have (like SPIDER [Frank 2006]) apply standard libraries for performing the analysis. These standard eigenvector-eigenvalue routines normally calculate all eigenvectors and eigenvalues of the system, and those are algorithms that scale proportional to N^2 (or worse), implying that the analysis of data sets of more than a few thousand elements will easily exceed the memory/computing capacity of any modern computer. With the poor scaling of the algorithms it can be difficult to analyse large data sets, a fact that is not generally appreciated in the statistical literature (where data sets are often very small by EM standards). Even in the specific EM literature, however, this issue is simply not appreciated (see, for example, paragraph 2.5 of [Frank, 2006]). Therefore, in times where datasets of more than 100,000 images are commonplace, one normally sees MSA applications limited to just a few thousand images (or 3D volumes).

Another issue that has been ignored in the literature is that the Chi-square metric of correspondence analysis is not appropriate for electron-microscopical phase-contrast data. Historically it happened to be the first MSA technique used because it was available (see appendix). However, the χ^2-metric of strict correspondence analysis uses is designed for positive data (like counting occurrences in histograms). To make the EM phase-contrast data positive one could threshold the image data to positive values only. This thresholding, however, is not justifiable for general signals and - at the very least - leads to wasting of half of the information for zero-average-density measurements. Alternatively, the negative values can be rendered positive by adding a constant to the data. This too has far-reaching consequences. The strong negative densities will end up as small positive densities and will

have very little contribution to the total variance of the data set, whereas the large positive values will become very large positive values with a disproportionally strong contribution to the total variance. For high-resolution EM work, however, the χ^2-metric must be considered obsolete since the publication of [Borland & van Heel 1990]. The metrics that are permissible for general signal processing are the Euclidean distance of principal components analysis and the modulation distance of modulation analysis discussed above.

New cryo-EM developments allowing the separation of subsets of 3D structures are emerging that have the potential to change the field of structural biology. Atomic resolution structures (~3Å) have hitherto been elucidated mainly by X-ray crystallography, where the biological molecules are confined to the rigidity of a crystal. In X-ray crystallography one collects data that *per definition* has been averaged over all unit cells of the crystal. For a better understanding of a biological process, however, it is essential to see the sequence of conformational changes and interactions a molecule or complex undergoes during its functional cycle. A sequence of 3D structures or "4D data analysis" is required. Single-particle cryo-EM provides a direct window into the solution, revealing different views of different complexes, in different functional states. The challenge is to extract all information from the noisy molecular images and to bring the existing structural differences to statistical significance. Revealing the full biological complexity of the data produced by 4D cryo-EM at atomic resolution is a major challenge in modern biology.

16. Conclusions

New developments in MSA eigenvector-eigenvalue data-compression approaches are changing the possibilities of single-particle cryo-EM. Some three decades after its first introduction to electron microscopy, MSA approaches are more alive than ever and they are being tailored for many new tasks. With the orders-of-magnitude speed increase achieved by the MPI parallelisation of the algorithms, a revival of the use of the technique is anticipated. One of the most challenging uses of the parallel MSA algorithm in the separation of heterogeneous data sets into different 3D structures associated with the different functional states of the macromolecular complexes we seek to understand.

17. Acknowledgements

First let us again point to the first MSA ideas that were inspired by Jean-Pierre Bretaudière (see appendix). This document reflects decades of development and experience with the MSA techniques discussed. During this period many individuals contributed to the various MSA projects (and not just the current authors) including: George Harauz, Lisa Borland, Ralf Schmidt, Elena Orlova, and Bruno Klaholz, to name a few (with apologies to those not acknowledged explicitly). Countless other scientists who used the programs gave us constructive feedback, reported bugs or other inconsistencies, and thus forced us to fine tune the programs and/or our ideas about the MSA methodologies and thus contributed indirectly. This document is the most recent of a series of papers including the full mathematics of the MSA, which series started with an unpublished lecture series on MSA approaches held by one of us (MvH) at the NY State Department of Health in Albany New York in early 1983. Last but not least we are grateful to Lena Orlova for pushing us so hard and for allowing us to submit this document well over the *absolute and final* deadline.

***NOTE*: this document is an electronic document and may be modified. Contact the authors for possible updates. Links to updates will be published on (www. single-particles.org).**

References

Adrian M, Dubochet J, Lepault J, & McDowall AW. **Cryo-electron microscopy of viruses**. *Nature* **308 (**1984) 32-36.

Benzécri J.-P. , **L'Analyse des Données Vol 2** (1973-1980) Dunod Paris.

Bock HH, **Origins and extensions of the *k*-means algorithm in cluster analysis**. *Journ@l Electronique d'Histoire des probabilités et de la Statistique* **4** (2008).

Borland L, van Heel M: **Classification of image data in conjugate representation spaces**. *J. Opt. Soc. Am.* 1990, **A7**: 601-610.

Boekema EJ, Berden JA, van Heel M: **Structure of mitochondrial F1-ATPase studied by electron microscopy and image processing**. *Bioch. Biophys. Acta* **851 (**1986) 353-360.

Clint M, Jennings A: **The evaluation of eigenvalues and eigenvectors of real symmetric matrices by simultaneous iteration**. *The Computer Journal* **13** (1970) 76-80.

Crowther RA: **Procedures for three-dimensional reconstruction of spherical viruses by Fourier synthesis from electron micrographs**. *Phil. Trans. Roy. Soc. Lond.* **B 261 (**1971) 221-230.

Crowther RA, Amos LA: **Harmonic analysis of electron images with rotational symmetry**. *J. Mol. Biol.* **60** (1971) 123-130.

DeRosier DJ, Klug A: **Reconstruction of three-dimensional structures from electron micrographs**. *Nature* **217 (**1968) 130-134.

Diday E: **Une nouvelle méthode en classification automatique et reconnaissance des formes la méthode des nuées dynamiques**. *Revue de Statistique Appliquée***19** (1971) 19-33.

Dube P, Tavares P, Lurz R, van Heel M: **Bacteriophage SPP1 portal protein: a DNA pump with 13-fold symmetry**. *EMBO J.* **15** (1993) 1303-1309.

Frank J: Three-Dimensional Electron Microscopy of Macromolecular Assemblies, Oxford University Press (2006).

Golub G, van Loan C: **Matrix computations**, third edition, The Johns Hopkins University Press Ltd. (1996) London.

Hoppe W, Gaßmann J, Hunsmann N, Schramm HJ, Sturm M: **Three-dimensional reconstruction of individual negatively stained yeast fatty acid synthetase molecules**

from tilt-series in the electron microscope. *Hoppe-Seyler's Z. Physiol. Chem.* **355 (**1974) 1483-1487.

Kastner B, Stöffler-Meilicke M, and Stöffler G. **Arrangement of the subunits in the ribosome of Escherichia coli: demonstration by immunoelectron microscopy**. *Proc Natl Acad Sci USA* **78** (1981) 6652–6656.
Klaholz BP, Myasnikov AG, van Heel M. **Release factor 3 seen on the ribosome during termination of protein synthesis.** *Nature* **427** (2004) 862-865.

Lake JA: **Ribosome structure determined by electron microscopy of Escherichia coli small subunits, large subunits and monomeric ribosomes.** *J Mol Biol* **105** (1976) 131-159.

Lebart L, Morineau A, et Tabard N, **Techniques de la Description Statistique** (1977) Dunod, Paris.

Lebart L., Morineau A., and Warwick K.M., **Multivariate Descriptive Statistical Analysis** (1984) Wiley, New York.

Leschziner AE, Nogales E: **Visualizing flexibility at molecular resolution: analysis of heterogeneity in single-particle electron microscopy reconstructions.** *Annu. Rev. Biophys. Biomol. Struct.* **36** (2007) 43-62.

Morineau, A., Lebart, L. **Specific clustering algorithms for large data sets and implementation in SPAD software**, **in**: **Classification as a tool of Research**, ed. Gaul, W., and Schager, M., **(1986)** North Holland Publishing Co.

Palade, GF: **A small particulate component of the cytoplasm.** *J. Biophys. Biochem. Cytol.* **1** (1955) 59–68.

Stewart A, Grigorieff N: **Noise bias in the refinement of structures derived from single particles.** *Ultramicroscopy***. 102 (**2004**)** 67-84.

Unwin PNT, Henderson R: **Molecular structure determination by electron microscopy of unstained crystalline specimens**. *J. Mol. Biol*., **94** (1975) 425-440.

van Bruggen EFJ, Wiebenga EH, Gruber M, **Structure and properties of hemocyanins. I. Electron micrographs of hemocyanin and apohemocyanin from Helix pomatia at different pH values**. *J Mol Biol*. **4** (1962a) 1-7.

van Bruggen EFJ, Wiebenga EH, Gruber M, **Structure and properties of hemocyanins. II. Electron micrographs of the hemocyanins of Sepia officinalis, Octopus vulgaris and Cancer pagurus**. *J Mol Biol* **4** (1962b) 8-9.

van Heel M, Frank J: **Classification of particles in noisy electron micrographs using correspondence analysis**, in: Pattern Recognition in Practice, Ed. E.S. Gelsema and L.N. Kanal, North Holland (1980) 235-243.

van Heel M, Frank J: **Use of multivariate statistics in analyzing the images of biological macromolecules**, *Ultramicroscopy* **6** (1981) 187-194.

van Heel M, Keegstra W: **IMAGIC: A fast, flexible and friendly image analysis software system.** *Ultramicroscopy* **7** (1981) 113-130.

van Heel M, Stöffler-Meilicke M: **Classification of images of the 30S *E. coli* ribosomal subunit**, in: *Proc 10th Intern. Cong. on Electron Microscopy, Hamburg* **3** (1982) 107-108.

van Heel M: **Multivariate Statistical Classification of Noisy Images (Randomly Oriented Biological Macromolecules)**, *Ultramicroscopy* **13** (1984a,)165-183.

van Heel M: **Three-dimensional reconstructions from projections with unknown angular relationship**. Proc. 8th Eur. Cong. on EM 1984b, Budapest, Vol.2, 1347-1348.

van Heel M, Stöffler-Meilicke M: **The characteristic views of E. coli and B. stearothermophilus 30S ribosomal subunits in the Electron Microscope**, *EMBO J.*, **4** (1985) 2389-2395.

Harauz G, van Heel M, **Exact filters for general geometry three dimensional reconstruction**, *Optik* **73** (1986) 146-156.

van Heel M: **Angular reconstitution: *a posteriori* assignment of projection directions for 3D reconstruction**. *Ultramicroscopy*, **21** (1987) 111-124.

van Heel M: **Classification of very large electron microscopical image data sets**, *Optik,* **82** (1989) 114-126.

van Heel M, Schatz M, Orlova EV: **Correlation functions revisite**d. *Ultramicroscopy* **46 (**1992) 304-316.

van Heel M, Harauz G, Orlova EV, Schmidt R, Schatz M: **A new generation of the IMAGIC image processing system**. *J. Struct. Biol.* **116** (1996) 17-24.

van Heel M, Schatz M: **Fourier Shell Correlation Threshold Criteria,** *J. Struct. Biol.* **151** (2005) 250-262.

Ward JH: **Hierarchical grouping to optimize an objective function**. *J. Amer. Statist. Assoc.* **58** (1982) 236-244.

White HE, Saibil HR, Ignatiou A, Orlova EV: **Recognition and separation of single particles with size variation by statistical analysis of their images**. *J. Mol. Biol.* **336** (2004) 453-460.

Zernike F: **Phase contrast, a new method for the microscopic observation of transparent objects, Part I**, *Physica* **9** (1942a) 686-693.

Zernike F: **Phase contrast, a new method for the microscopic observation of transparent objects, Part II**, *Physica* **9** (1942b) 974-986.

Appendix*

Jean-Pierre Bretaudière (1946-2008) and the early days of multivariate statistics in electron microscopy.

Marin van Heel,

Imperial College London

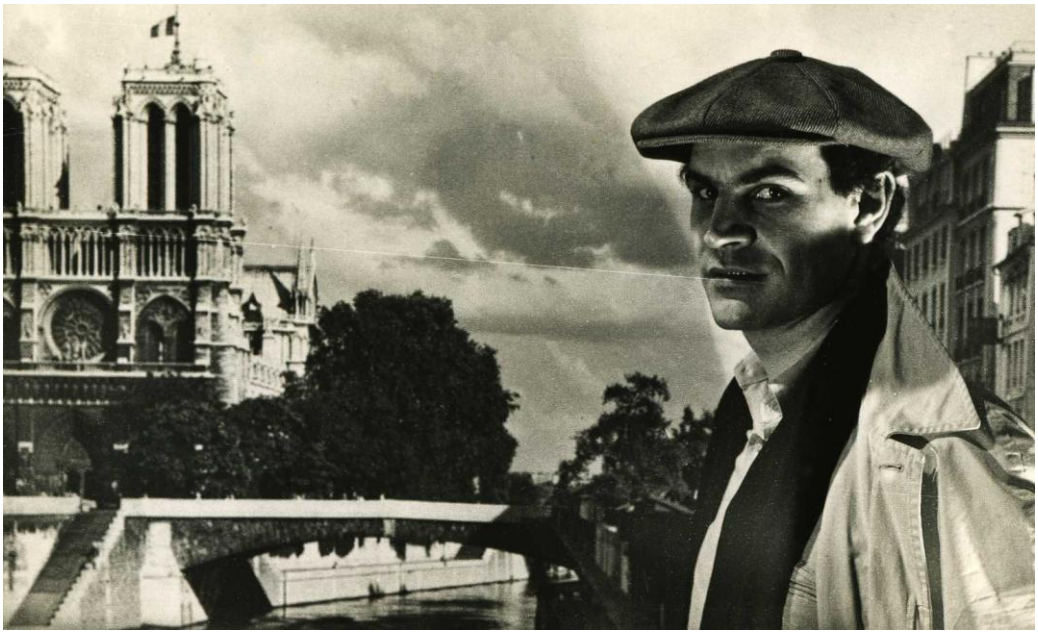

Fig A-1: **Jean-Pierre Bretaudière in front of a Paris poster – photo: Jan Galligan, Albany 1979**

Jean-Pierre Bretaudière had a very fundamental influence in the early days of multivariate statistical analysis of the images of single particles in electron microscopy, an influence that was not very well documented in terms of co-authorships or in other written documents. Let me explain how he was involved in the "intelligent averaging" of specific views of single particle images. Averaging of a great many "unit cells" of molecules arranged as a helical assembly [De Rosier & Moore 1970] or as a two-dimensional crystal [Unwin & Henderson 1975], had at the time made quite an impression in the field and many electron microscopy groups worldwide thus started image processing projects. The first ideas of applying the concept of averaging to the images of individual macromolecules had been formulated [Saxton & Frank 1977] and implemented in Owen Saxton's early IMPROC software system. It was not yet clear at the time how best to find the 3D structure of a macromolecular complex based on individual two-dimensional projection images of the molecules. Would single-particle tomography such as proposed by the group of Hoppe [Hoppe *et al.* 1974] be the way towards, or did one first have to find all the different views presented by a macromolecular complex? Around 1980, we were still a decade away from answering such questions.

*__Appendix to__: Van Heel M, Portugal R & Schatz M: **Multivariate Statistical Analysis in Single Particle (Cryo) Electron Microscopy**. In: "**An electronic text book: Electron microscopy in Life Science**", 3D-EM Network of Excellence, Editors: A. Verkley and E. Orlova (2009).

Since 1976, I had been employed as a project scientist in the electron microscopy group of Professor Ernst van Bruggen of the Biochemistry Department of the University of Groningen, The Netherlands. Erni van Bruggen had been most impressed by the early "averaging" successes of the group around Aaron Klug in Cambridge and wanted to incorporate those techniques in his research. It was my task to develop image processing software for data collected either with the transmission electron microscope (TEM) or the Scanning version of that instrument (a "STEM"). In the fall of 1979, I went to visit Albany, for a 6-week period, to work with Joachim Frank on automatically finding sub-populations of images in a heterogeneous population as discussed in the main paper. We had met two years earlier at an EMBO image processing course in Basel and we had been corresponding ever since. We both agreed that when averaging images of individual macromolecules, the presence of a mixture of different views was a serious and important complication: a challenging problem in need of a solution. I travelled to Albany taking with me two different sets of images in which different views could already be visually distinguished. One set consisted of images of worm hemoglobin (two different preparations: one with a clearly visible extra subunit in the middle and one without) and the other dataset of arthropod 4x6 hemocyanins where the difference between "flip" and "flop" could be directly seen (**Fig A-2**).

What followed were many nights in the windowless catacombs of the New York State Department of Health (Division Laboratories and Research, now the Wadsworth Centre) beneath the Empire State Plaza. During these late sessions I met Jean-Pierre Bretaudière ("JP") while working at the same PDP-11 45 computer (**Fig A-3/A-4/A-5**). We first talked about other issues than science, such as his personal ties to Paris (**Fig A-1**) where I had spent almost all of 1971 as a student. We also discussed our actual work and I explained the first attempts Joachim and I were making at separating subsets of images from the overall population of molecular images. Jean-Pierre explained what he was doing at the State Department of Health: quality control of medical laboratories in the state of New York. The Health Department would send out identical blood samples to many different medical laboratories. The results of the various chemical tests from the different labs were then compared using a French multivariate statistical technique called "l'analyse des correspondances" [Bretaudière, 1981]. At the time I had never heard of multivariate statistical analysis ("MSA"), let alone of Benzécri's "correspondence analysis" [Benzécri, 1973-1980].

Jean-Pierre was immediately convinced that his "*AFC*" program (for: "Analyse Factorielle des Correspondances") would be able to help solve our problem and handle our "massive" amount of image data. The *AFC* program was a modified version of the correspondence analysis program *"CORAN"* by Jean Paul Benzécri [Benzécri, 1973/1980]. What was "massive" in this context was a very relative and very dated concept: at the time we had less than 64 images of 64x64 pixels = 4096 pixels each; that is a total data set size of less than one megabyte (each image being 16 Kbyte in floating point format). An uncompressed picture from even the cheapest digital camera today is much larger than one megabyte! However, it was characteristic for those early days in computing, that the 16-bit PDP-11 computer would only address 2^{16} bytes = 64 Kbytes. It was thus difficult to fit any "major" computational problem in the central memory of such a computer! The critical memory requirement in our case was the "core" needed for the square variance-covariance matrix (see main article) which, for 64 images would already occupy 16 Kbyte of the precious available address space.

As we sat together with Joachim the following morning to discuss the matter, we were all convinced that this was very much worth trying. Joachim wrote a small conversion program to allow JP's AFC program to read the aligned images produced by the SPIDER program. The result of this first correspondence analysis of single particles was an instant success!

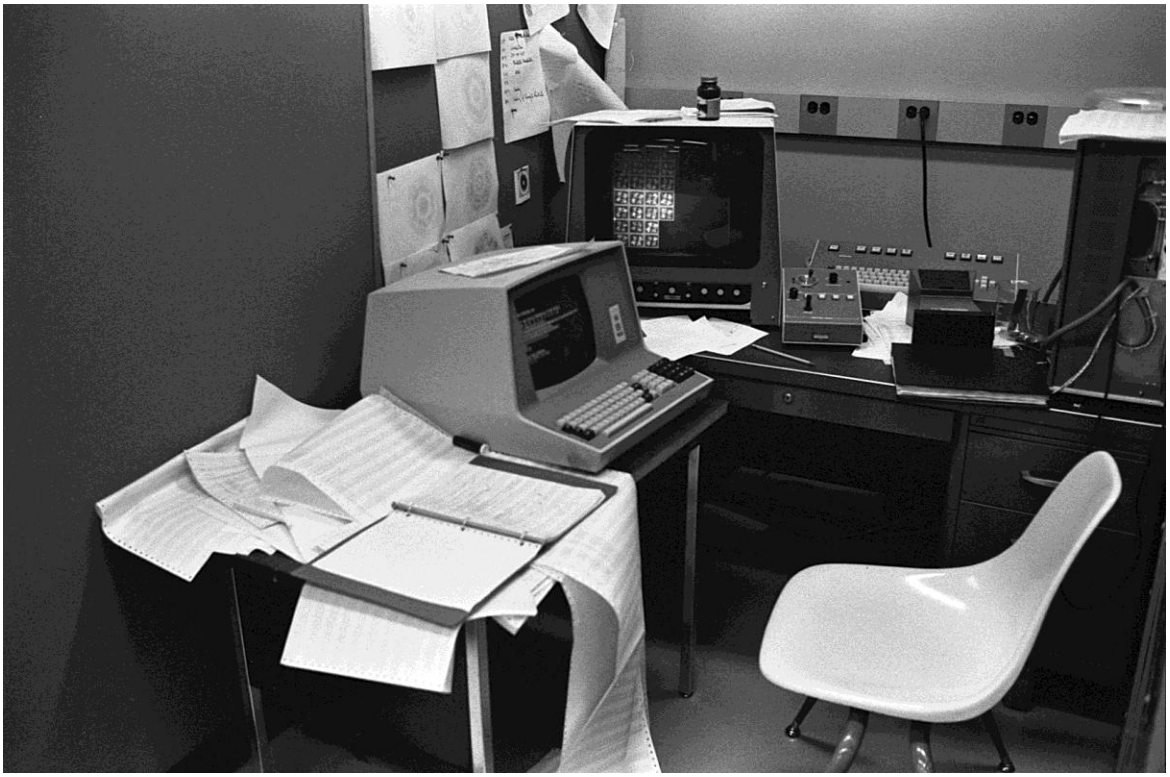

Fig A-2: Wadsworth Computer Centre, 1979. Visible on the image monitor (controlled by the 24x80 characters VDU) is a gallery of 4x6 arthropod hemocyanin images used for the first correspondence analysis of EM images. Already in this gallery one manages to directly see the difference between the "flip" and "flop" versions of the 4x6 half molecules of the *Limulus polyphemus* hemocyanin – photo: Marin van Heel, Albany 1979.

The first trial was on a giant annelid hemoglobin data set extracted from images of hemoglobins from two different species. The giant annelid hemoglobin of *Oenone fulgida* visibly had extra density in the centre [Van Bruggen & Weber, 1974] compared to the well known giant hemoglobin of the common earthworm *Lumbricus terrestris* which rather had a "hole" in the centre of its double-ring geometry. The separation of the two species, artificially mixed for this analysis, was so obvious that we immediately moved to a single data set with real internal variations. We thus moved to the second data set I had brought from Groningen, negative-stain images of *Limulus polyphemus* 4x6-meric hemocyanins (a dissociation product of the full 8x6-meric native hemocyanin).

On the correspondence analysis maps of these images four clear groups ("classes") were visible. These reflected the "flip and flop" versions of the 4x6 oligomer where the right hand 2x6 part can be shifted up or down with respect to the left-hand side 2x6-mer. This behaviour had already been anticipated based on visual inspection of the images (**Fig A-2**). However, that effect would have led to a subdivision into two groups rather than four. The other effect, which I had not anticipated, split both the "flip" and the "flop" groups in two, namely that of "rocking". The four hexamers constituting the 4x6-mer were apparently not in one plane

which led to two stable "rocking" positions where 3 out of 4 hexamers would be in direct contact with the carbon support film. This explanation, however, was not so straightforward because the alignment algorithm used at that time would itself introduce a 180º ambiguity in the in-plane rotational alignment. Both the shape of the molecule and the flaws of the alignment scheme thus led to the same effect. This issue was eventually resolved after confusing argumentations with referees [van Heel & Frank 1980; van Heel & Frank 1981]. The books that JP recommended [Benzécri 1973/1980]; and especially [Lebart *et al.* 1977] became my "bibles" for years to come (English versions of these books have since appeared making that literature more accessible).

In these early MSA results, we would print out two-dimensional maps of the positions of the images on the first factorial co-ordinates and then draw a circle around a cluster of images, call that a "class" and then sum the members of that class for further interpretation. This visual classification, after eigenvector-eigenvalue data reduction, obviously had its shortcomings and was impossible to perform when more than ~3 factorial axes were to be taken into account. It was JP who also pointed me in the direction of automatic classification and the wealth of literature on the subject [van Heel, Bretaudière & Frank, 1982], which then led to the introduction of automatic Hierarchical Ascendant Classification, "HCA", in electron microscopy [van Heel, 1984].

Fig A-3: Wadsworth Computer Centre, 1979. The PDP-11/45 "minicomputer" – photo: Marin van Heel, Albany 1979

Fig A-4: Jean-Pierre Bretaudière in the Wadsworth Computer Centre – photo: Robert Rej, Albany 1981.

Jean-Pierre was a wild guy - bursting with energy, full of ideas and projects, always burning the candle at both ends. I have vivid memories of, after a late evening session at work and an even longer night in a bar, stumbling out of the bar roaring with laughter along with JP and Vicky, Joachim's programmer who later became JP's wife. JP was always exuberant in his love for life and all its adventures. Jean-Pierre, having seen the success of correspondence analysis in single-particle electron microscopy, became directly interested in EM image processing. He changed fields and moved to the University of Texas Medical Center in Houston, Texas, where he developed the SUPRIM image processing system for electron microscopy [Schroeter & Bretaudière, 1996]. In Houston, being JP, he was proud to declare himself more Texan than the Texans: he bought a ten-gallon hat, a huge Lincoln with "BRETO" vanity plates, and a large gun to defend his home and family. I still vividly remember the self-satisfied smile on his face when he told me this...

The last time we spoke, already many years ago, was about the time he left Houston to return to France. We spoke of his health issues and about his plans to leave the single-particle EM field - again. JP was never afraid of rigorous decisions. To this day, I am sorry we did not include JP as a co-author of the first papers that resulted [van Heel & Frank 1980; van Heel & Frank 1981]. He was offered co-authorship but refused. Maybe Joachim and I simply did not push him hard enough. His co-authorship would have reflected his essential early contributions to the ideas in this field. The only printed evidence of our early collaboration which remains is the abstract for the European electron microscopy meeting in Hamburg in 1982 mentioned above [Van Heel, Bretaudière & Frank, 1982].

The beautiful portrait of JP by Jan Galligan (**Fig A-1**), which so clearly documents his *joie de vivre*, resurfaced in my archives a few years ago. After scanning it, I searched the web to find JP's current email address hoping to put a smile on his face. Unfortunately I failed to find a current address. Only recently, after receiving the sad message of Jean-Pierre's death, did I learn he had moved to Madagascar, after selling his very successful French computer business – Brett Computers – moving away from western society's rat-race. Some say that scientists are dry and boring people… There is no doubt that with JP no longer around, the world of science has become a slightly more boring small universe.

I am grateful to Jan Galligan and Robert Rej for letting me to use their pictures, and for their support and constructive suggestions.

References

Benzécri J.-P. **L'Analyse des Données Vol 2** (1973-1980) Dunod Paris.

Bretaudière JP, Dumont G, Rej R, Bailly : **Suitability of control materials. General principles and methods of investigation**. *Clin. Chem,* **27/6** (1981) 798-805,

DeRosier DJ, Moore PB: **Reconstruction of three-dimensional images from electron micrographs of structures with helical symmetry**. *J. Mol. Biol.* **52**: (1970) 355-369.

Hoppe W, Gaßmann J, Hunsmann N, Schramm HJ, Sturm M: **Three-dimensional reconstruction of individual negatively stained yeast fatty acid synthetase molecules from tilt-series in the electron microscope**. *Hoppe-Seyler's Z. Physiol. Chem.* 1974, **355**: 1483-1487.

Lebart L, Morineau A, et Tabard N, **Techniques de la Description Statistique** (1977)

Saxton WO, Frank J: **Motif detection in quantum noise-limited electron micrographs by cross-correlation**. *Ultramicroscopy* **2** (1977) 219-227.

Schroeter JP & Bretaudière JP: **SUPRIM: Easily modified image processing software**: *J. Struct. Biol.*116 (1996) 131-137.

Unwin PNT & Henderson R: **Molecular structure determination by electron microscopy of unstained crystalline specimens**. *J. Mol. Biol*. **94** (1975) 425-440.

Van Bruggen EFJ, Weber RE: **Erythrocruorin with anomalous quaternary structure from the polychaete *Oenone fulgida*.** Biochim. Biophys. Acta 359 (1974) 210-214.

Van Heel M, Frank J: **Classification of particles in noisy electron micrographs using correspondence analysis**, in: Pattern Recognition in Practice, Ed. E.S. Gelsema and L.N. Kanal, North Holland (1980) 235-243.

Van Heel M, Frank J: **Use of multivariate statistics in analyzing the images of biological macromolecules**, *Ultramicroscopy* **6** (1981) 187-194.

Van Heel M, Bretaudière JP, and Frank J: **Classification and Multi-Reference Alignment of Images of Macromolecules**, in: Proc. 10th Intern. Congress on Electron Microscopy, Hamburg, 1982, Vol. 1, pp. 563-564.

Van Heel M: **Multivariate Statistical Classification of Noisy Images (Randomly Oriented Biological Macromolecules)**, *Ultramicroscopy* **13** (1984) 165-183.