# **Singular Value Decomposition (SVD) of microarray data.**

In the previous chapter we have learned the SVD decomposition of any *m* x *n* matrix ***A*** as:

*m* x *n* *m* x *m* *m* x *n* *n* x *n*

such that:

with:

An extremely important application of this concept is in *systems biology*. Consider the case of microarray data ***X***, in which *xij* is the expression level of the *i*th gene in the *j*th assay. The elements of the *i*th row of ***X***form the *n*-dimensional vector **g***i*, which we refer to as the *transcriptional response* of the *i*th gene. Alternatively, the elements of the *j*th column of ***X***form the *m*-dimensional vector **a***j*, which we refer to as the *expression profile* of the *j*th assay.

***n*** *assays*

**g1** = transcriptional response of the 1st **g**ene

***m*** *genes*

**a1** =expression profile of the 1st **a**ssay

If one conditions the data matrix ***X***by *centering* each column, then ***X*T*X***(*n x n*) is proportional to the covariance matrix of the variables **a***j* (*i.e.*, the covariance matrix of the expression profiles). *Eigen* decomposition of ***X*T*X***yields ***V***: so, the right singular vectors ***v1***, ***v2***, ... are the *principal components axes* (an orthonormal basis) of the *row space* {**g***i*}, the space of the gene transcriptional responses.

If one conditions the data matrix ***X***by *centering* each row, ***XX*T** (*m x m*) is proportional to the covariance matrix of the variables **g***j* (*i.e.* the covariance matrix of the genes transcriptional responses). In this case, the left singular vectors ***u1***, ***u2***, ... are the *principal components axes* (an orthonormal basis) of the *column space* {**a***j*}, the space of the assay expression profiles.

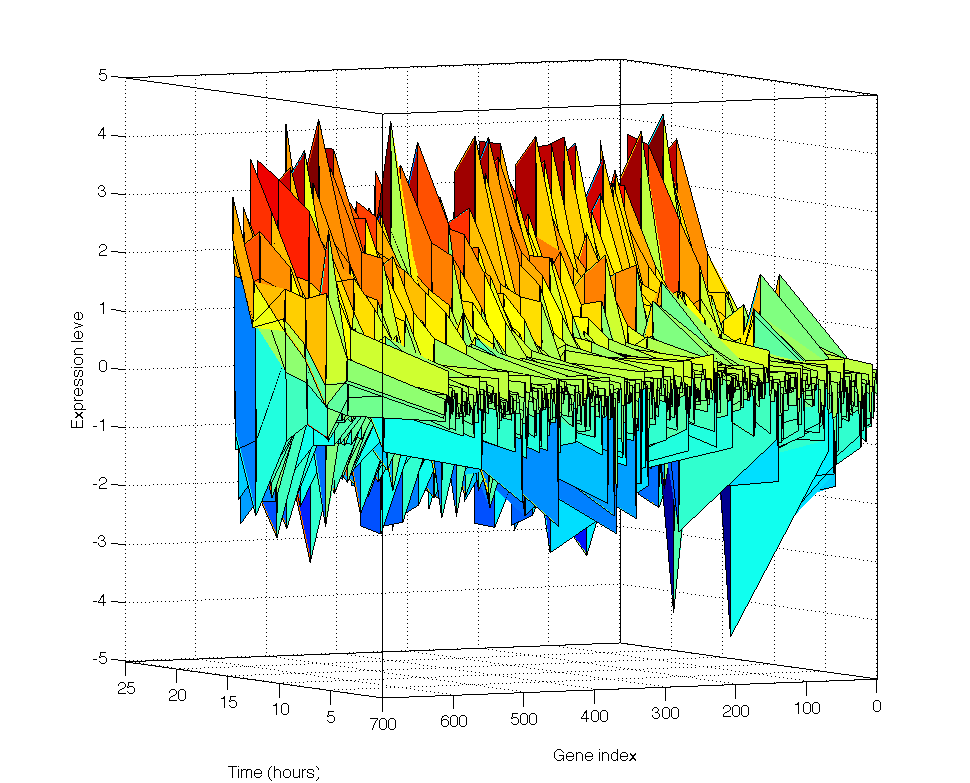
Thus, if we have a gene expression data matrix ***X***with *n* columns corresponding to assays, and *m* rows corresponding to genes, the **SVD** of ***X***produces two orthonormal bases of right and left singular vectors. It is a widely used convention to refer to the left singular vectors ***u****k* as *eigenassays* (or *eigenprofiles*) and to the right singular vectors ***v****k* as *eigengenes*. Furthermore, when the number of genes *m* is larger than the number of assays *n* (the most common case), usually the '*economy*' SVD is calculated in which the first *n*columns of ***U*** are included (regardless of the matrix rank) and therefore ***Σ*** and ***V***  are both *n* x *n*.

*m* x *n* *m* x *n* *n* x *n* *n* x *n*

The *systems biology* interpretation of the 'economy' **SVD** decomposition for microarray data is then:



Notice how the coefficients of every *eigenassay* are the contributions from the different **g***j* genes, and the coefficients of every *eigengene* are the contributions from the different **a***j* assays.

Let's see how this works in practice. As an example, we will consider microarray data in which the expression of *m*= 614 yeast genes was monitored for *n**=*7 different times (up to 20 hours) after changing the main nutrient in the culture media at time 0.

load filteredyeastdata

data = yeastvalues; [m,n] = size(data);

[XI,YI]=meshgrid(times,1:m);

figure;surf(XI,YI,data);

box('on'); ylim([0 614]); xlim([0 21]);

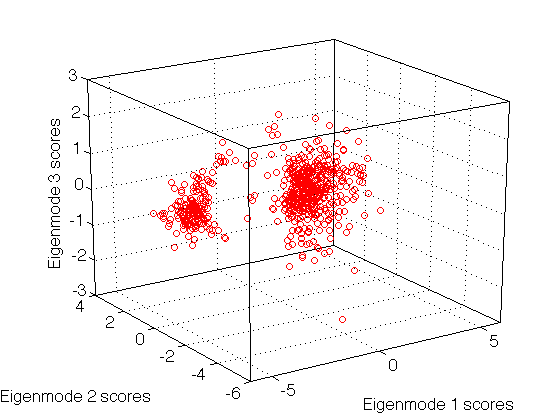
xlabel('Time (hours)'); ylabel('Gene index')

zlabel('Expression level')

We can see how the expression of some genes goes up and that of others goes down. We ask the question: **are there genes that have a common pattern of activation or inhibition?** In other words, we are looking at the *row space* {**g***i*}, the space of the gene transcriptional responses.

Since the rank of ***X*** is *r* = 7,intuitively we can see how the transcriptional response of each gene can be considered as a linear combination of 7 *eigengenes*, each *eigengene* being a vector of 7 elements, and each element representing some level of transcriptional response at that time. Therefore, we want to find the representation (the *scores*) of the transcriptional responses (the rows of ***X***)in terms of *eigengenes* (the orthogonal basis ***V*** for the *row space* of ***X***). In order to do this we need first to *transpose* ***X***, then apply a change of basis to ***V***, and finally transpose back:

Notice that since:

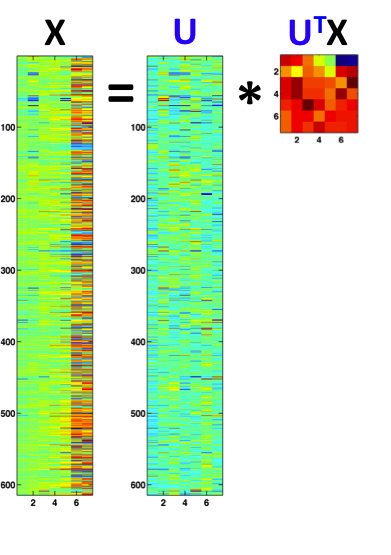


multiplying both side by ***V*** on the right:

[U,S,V] = svd(data,'econ');

XV = data\*V;US = U\*S;

figure; plot3(XV(:,1),XV(:,2),...

 XV(:,3),'ro')

xlabel('Eigengene 1 scores ')

ylabel('Eigengene 2 scores ')

zlabel('Eigengene 3 scores ')

xlim([-6 6]),grid('on'),box('on')

Thus, the rows of the product matrix are the coordinates (*scores*)of the transcriptional responses of each gene in the 7 principal components (*eigengenes)* ***v****k* of the *row space* {***gi***} of ***X***. It follows that if we plot the first 3 columns of the matrix against each other, genes that have transcriptional responses with similar contributions from the top 3 *eigengenes* will have similar coordinates and thus will appear clustered. This type of plot can be considered as a projection of the entire genome transcriptional response onto the first 3 *eigengenes* (out of a total of 7 *eigengenes*).

scores

loadings

In a similar way we can operate on multiplying both side by ***UT*** on the left:

Thus, the columns of the matrix contain the coordinates (*the scores*) of each assay in the 7 principal components (*eigenassays)* ***u****k* of the *column space* {***ai***} of ***X***. The biological meaning of the *left singular vectors* ***u****k*  (*the loadings*) is clarified by considering the SVD decomposition as a two terms product:

Notice, this is nothing other than a PCA of ***X***:

***T*** = score matrix

***P*** = loading matrix

UX = U'\*data;

figure;imagesc(data);figure;imagesc(U);figure;imagesc(UX);

This linear decomposition represents a model of gene expression based on the assumption that the expression profiles (the columns in ***X***) are determined by a combination of different ‘expression normal modes' (the columns of ***U***). The combinations of these modes giving origin to each column of ***X*** are in the corresponding column of . Each *expression normal mode* reveals groups of genes that are activated or inhibited in a coordinated way possibly because they belong to a metabolic pathway or are involved in a specific cell function. Since the sign of the singular vectors is arbitrary, if we want to have a consistent result we can change the sign so that the mean is higher than the median:

for n=1:n

vec\_sign = sign(mean(U(:,n))-median(U(:,n)));

U(:,n)=vec\_sign\*U(:,n);V(:,n)=vec\_sign\*V(:,n);

end

In conclusion, the rationale behind using the SVD remains very simple:

***U*** provides an orthonormal basis for the column space of ***X***.

***V*** provides an orthonormal basis for the row space of ***X***.

Therefore, to represent:

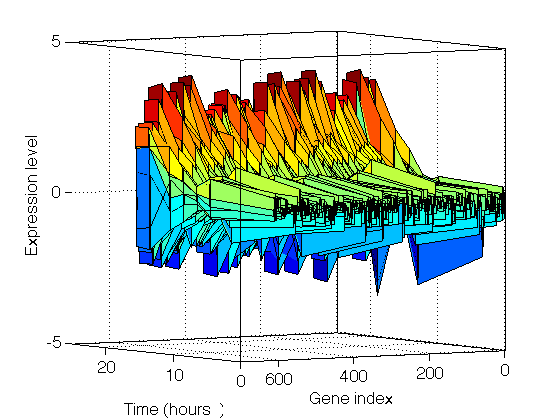
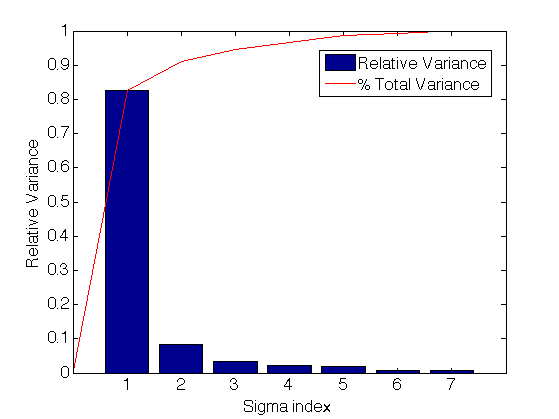
***X*** in an orthonormal basis of its column space calculate (***n****x****n***)*.* Optionally center *the* ***rows*** of ***X***as for calculating cov = ***XXT***

**⇒** subtract the mean of all columns from each column.

***X*** in an orthonormal basis of its row space calculate (***m****x****n***)*.* Optionally center *the* ***columns*** of ***X***as for calculating cov = ***XTX****.*

**⇒** subtract the mean of all rows from each row.

There is additional information that can be obtained from the **SVD** of the data matrix ***X***. We can look at a 'scree' plot of the Σ2 values to determine the percentage of variance *explained* by the top *singular vectors*.



D = diag(S).^2;

sumD = cumsum(D);

E = sumD/sumD(end);

relD = D/sumD(end);

figure;bar(relD);hold on

plot([0:7],[0 ; E],'-r')

xlabel('Sigma index')

ylabel('Relative Variance ')

legend('Relative Variance ','% Total Variance','Location','Best')

In this case over 90% of the total variance in trascriptional response/expression profile is explained by the top 2 eigengene/eigenassay combinations. Therefore we could re-represent our microarray data using only those combinations:

.

data\_red = U(:,1:2)\*S(1:2,1:2)\*V(:,1:2)';

[XI,YI]=meshgrid(times,1:m);

figure;surf(XI,YI,data\_red);

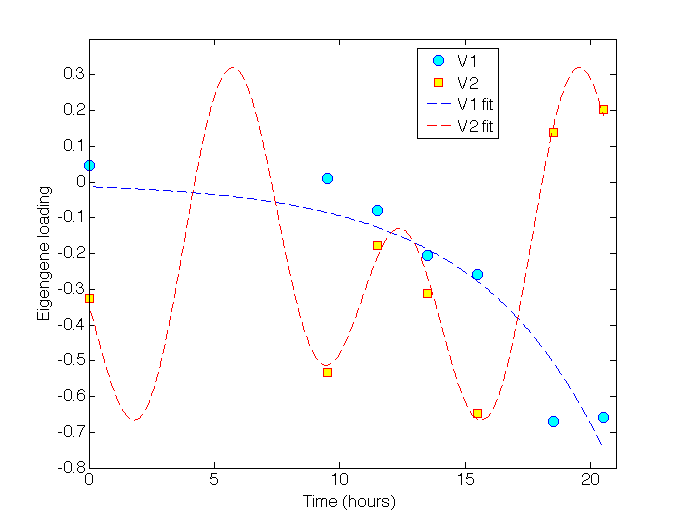
box('on'); xlabel('Time (hours )')

ylabel('Gene index ')

zlabel('Expression level ')

xlim([0 25]),ylim([0 700]),zlim([-5 5])

***Xred(1,2)*** is the best rank-2 approximation of ***X***, which was originally a rank-7 matrix.

When assays correspond to samplings of an ordinal or continuous variable (*e.g.*, time; radiation dose; drug concentration), a plot of the elements of the *eigengenes* ***v****k* can reveal recognizable patterns. In our example, the first two *eigengenes* show respectively an exponential structure with a rate constant of ~0.2/hr and a cyclic structure with two superimposed sine waves that cover also a missing time point around 5 hours. However, it is important to notice that neither *eigengene* is exactly like the underlying sine or exponential pattern, because sine waves and exponential patterns cannot simultaneously be *right singular vectors*, as they are not orthogonal.

V1 = V(:,1);V2 = V(:,2);figure; plot(times, V1,'ob',times,V2,'sr');hold on

f = fittype('a\*exp(b\*x)');[Expon,GOF] = fit(times',V1,f,'StartPoint',[-.01 .2]);

U1 = coeffvalues(Expon)

xvec = [0:0.1:20.5];

yvec = Expon(xvec);

plot(xvec,yvec,'--b')

f = ...

fittype('a1\*sin(x\*b1 + c1) + a2\*sin(x\*b2 + c2)');

[Cyclic,GOF] = ...

fit(times',V2,f,'StartPoint',[0.5 0 0 0.5 0 0]);

U2 = coeffvalues(Cyclic)

xvec = [0:0.1:20.5];

yvec = Cyclic(xvec);

plot(xvec,yvec,'--r')

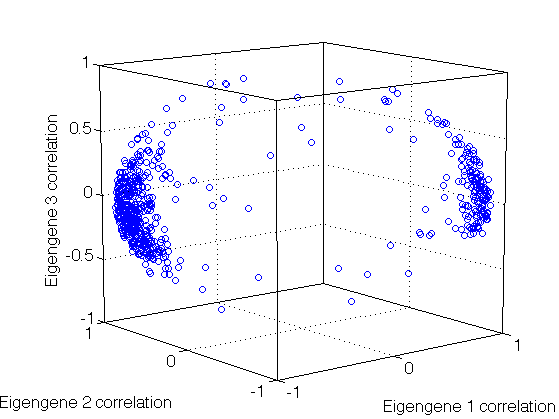
xlabel('Time (hours) ')

ylabel('Eigengene loading')

xlim([0 21]);ylim([-1 0.5])

legend('V1','V2','V1 fit','V2 fit','Location','Best')

A final type of analysis afforded by the SVD is the generation of *correlation plots*. First we calculate the matrix of Pearson correlation coefficients (the correlation matrix) of each gene’s transcriptional response with each of the *eigengenes*: we recall that this is the cosine of the angle between the centered transcriptional responses and the centered *eigengenes*. The correlation matrix has dimensions **614** x **7**, and each column holds the correlation of all the genes transcriptional responses with a given *eigengene*.

corrGV = corr(data',V);

plot3(corrGV(:,1),corrGV(:,2),corrGV(:,3),'ob')

xlabel('Eigengene 1 correlation ')

ylabel('Eigengene 2 correlation ')

zlabel('Eigengene 3 correlation ')

grid('on'); box('on')

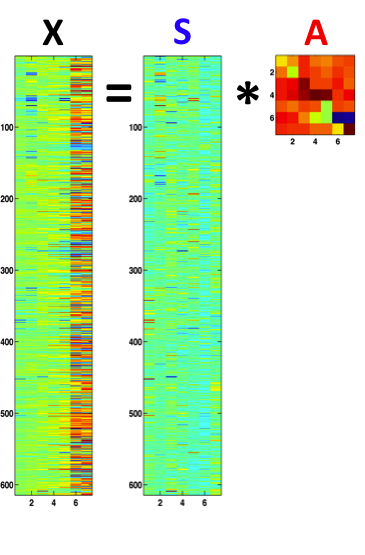
Next, like we did for the projection analysis, we plot the first 3 column vectors against each other: genes that have similar transcriptional response correlations to all three *eigengenes* will appear clustered.

Both the previous projection plots and the new correlation plot show clearly that the 614 yeast genes surveyed in this microarray study cluster in two distinct groups with respect to their transcriptional response. The two clusters correspond to the two types of transcriptional response observed (exponential or sinusoidal). The correlation plot shows clearly that the two types of response are anticorrelated: genes with strong exponential pattern have very small sinusoidal pattern, and genes with strong sinusoidal pattern of expression have very small exponential contribution. The correlation plot also show that both groups of genes spread further out with respect to the 3rd *eigengene* without separating into additional new clusters.

**SPECIAL TOPIC: Microarray analysis by Independent Component Analysis.**

We have seen how SVD can be used to decompose the space of the transcriptional responses {**g***i*} into linear combinations of *eigengenes*, and the space of the expression profiles {**a***j*} into linear combinations of *eigenassays*. Each *eigenassay* is an *'expression normal mode'* of the genome, with all modes orthogonal to each other and decorrelated.However, we have seen that decorrelation between variables does not automatically garantee their independence. The identification of modes that are not only decorrelated, but also independent is the goal of *Independent Component Analysis* (**ICA**). ICA is widely used for blind source separation and denoising: in the case of microarray data ICA is used to transform linearly the expression profiles into components with minimal statistical dependencies between them. The interpretation of this analysis is similar to that afforded by SVD: the independent components reflect the *influence* of *unobserved variables* that control the expression of all the genes. Thus, each component defines particular levels of induction or repression of individual genes as *'independent modes of expression*' of the genome. Very often the dominant modes can be related to the activation or repression of a particular cellular function.

In standard ICA analysis the *observed components* ***X*** can be derived from the *independent components* ***S*** by means of a *mixing matrix* ***A***. Traditionally in ICA both the observed and the independent components are represented as *row vectors*, and the ICA solution has the form:

However, microarray data is typically in the form of *m*genes and *n*samples, and thus the *observed components* ***X*** are usually represented as *column vectors*. Therefore, the ICA analysis typically starts with *transposing* the microarray data ***Xcols*** to obtain the solution:

The solution is then converted to standard microarray format by *transposing* again:

This format has the independent components ***S*** and the mixing matrix ***A*** switched in positions with respect to the traditional ICA expression (***X = AS***):

load filteredyeastdata

data = yeastvalues;

[m,n] = size(data);

% [S1, A1, W1] = fastica(data')

[S1,W1] = RADICAL(data');

A1 = inv(W1);

S = S1';

A = A1';

covS = cov(S)

Since the sign of the independent components is arbitrary, if we want to have a consistent result we can change the sign so that the mean is higher than the median:

for n=1:n

vec\_sign =...

sign(mean(S(:,n))-median(S(:,n)))

S(:,n)=vec\_sign\*S(:,n);

A(n,:)=vec\_sign\*A(n,:);

end

X = S\*A;

figure;imagesc(X)

figure;imagesc(S)

figure;imagesc(A)

This linear decomposition represents a model of gene expression based on the assumption that the ***X*** expression profiles (samples) are determined by a combination of hidden regulatory variables producing different *‘expression independent modes'* (the columns of ***S***). The *coefficients* for the combinations of these modes giving origin to each ***k*** column in ***X*** are in the corresponding ***k*** column of ***A***. Each expression mode may reveal a particularly active state of groups of genes belonging to a metabolic pathway or specific cell function. This is very reminiscent of the SVD linear decomposition:

with ICA ***S*** and ***A*** corresponding to SVD ***U*** and ***UTX***. The fundamental difference between the two decompositions is that while the SVD ***U*** modes are *orthogonal* and therefore uncorrelated, the ICA modes are also *independent* in addition to being uncorrelated, with the known difference between *uncorrelatedness* and *independence*.

However, while in PCA/SVD *eigen/singular* vectors are ranked based on the magnitude of the corresponding *eigen/singular* values, the independent components identified by ICA are in no specific order. Thus, in order to select which ones to retain and which to discard to filter out noise in the data, two criteria are usually adopted:

It's worth recalling here that the specifc contribution of the ***k*** independent component to the entire data set can be calculated as the *dyadic* (outer) product (for which we use here the symbol to distinguish it from the*Kronecker tensor* product):

X\_1 = S(:,1)\*A(1,:);

so that the entire set can be recovered as the sum of all the *dyadic* products:

X\_all = zeros(m,n,n);

for i = 1:n

X\_all(:,:,i) = S(:,i)\*A(i,:);

end

figure;imagesc(X\_all(:,:,3))

figure;imagesc(sum(X\_all,3))

1. the fraction of total variance in the data associated with a ***k*** independent component (*column*) of ***S***. This is calculated as the variance ***JA*** of the corresponding ***k*** *row* of the mixing matrix ***A***.

J\_a = var(A,0,2)’

2. the *contrast* value ***JS*** of each independent component in ***S***.Thisis becausewhen compared to noise, the biological components should not only capture a higher amount of the data variance, but also be more *informative* and *less random*. A convenient measure of the information content of a variable is given by its *entropy* ***H***.

According to Shannon’s information theory the *entropy* ***H*** of a discrete random variable ***X*** with possible values {*x*1, …, *xn*} and *probability distribution function* *p*(***X***) can be defined as:

Here *E* is the expected value operator, and ***I*** is the *information content* of ***X***. The entropy of ***X*** can be calculated from the expression:

where *b* is the base of the logarithm used. Common values of *b* are 2, *e*, and 10. When *b* = 2, the units of entropy are commonly referred to as *bits*. For most applications, the probability is replaced with the observed frequency of or of a *bin* of values centered around in ***X***.

A fundamental result of information theory is that a continuous unbounded *gaussian* variable has the largest entropy among all unbounded random variables of equal variance. Thus, it is customary to use a *contrast* function based on the concept of *negative entropy* ***JS***, defined as the difference between the entropy of a purely *gaussian* variable and that of the variable under study:

gauss\_vec = random('normal',0, 1,[m,1]);

S\_tag = [S gauss\_vec];

S\_binned = S\_tag;

for j = 1:size(S\_tag,2)

[bins,edges] = internal.stats.histbins(S\_tag(:));

nbins = length(bins)

for i = 1:nbins

ind = S\_tag(:,j)>= edges(i) & S\_tag(:,j)<= edges(i+1);

S\_binned(ind,j) = i;

end

end

H\_s = Entropy(S\_binned)

J\_s = H\_s(end)-H\_s(1:end-1)

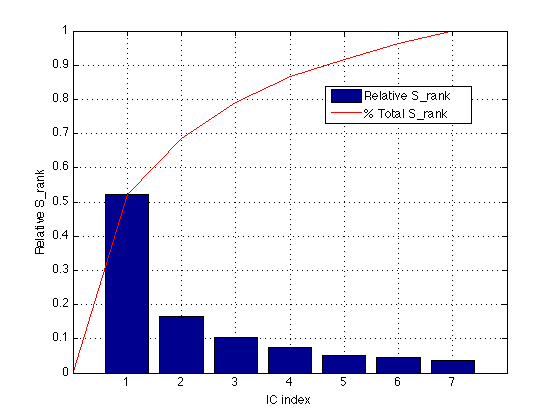
Then, to take both properties into account, and without considering a biological meaning behind the exact order, we sort the independent components according to a linear combination of both quantities, scaled by their mean values, with some arbitrary constant ***c*** between 0 and 1:

mean\_J\_s = mean(J\_s)

mean\_J\_a = mean(J\_a)

c = 0.5;

S\_rank = c\*J\_s/mean\_J\_s + (1-c)\*J\_a/mean\_J\_a

 [~,S\_rank\_ind] = sort(S\_rank,'descend')

We can make a 'scree' plot of the S\_rank to help us decide which IC to retain.

sumJ = cumsum(S\_rank(S\_rank\_ind));

E = sumJ/sumJ(end);

relJ = S\_rank(S\_rank\_ind)/sumJ(end);

figure;bar(relJ);

hold on

plot([0:7],[0 E],'-r')

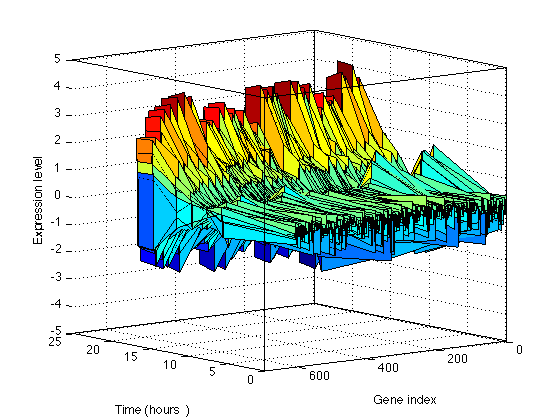
xlabel('IC index')

ylabel('Relative S\\_rank ')

legend('Relative S\\_rank ','% Total S\\_rank','Location','Best')

grid on

In this case the scree plot shows that almost 90% of all the information is contained in the top 4 ranked IC's. Based on this information, we can decide to take only a subset of IC’s to reconstitute the data. For example, to take only components 1,3,5,7 we would sum:

or we can simply take the top 4 ranked components identified with the scree plot:

data\_recov =...

S(:,S\_rank\_ind(1:4))\*...

A(S\_rank\_ind(1:4),:);

% figure;imagesc(data)

% figure;imagesc(data\_recov)

[XI,YI]=meshgrid(times,1:m);

figure;surf(XI,YI,data\_recov);

box('on'); xlabel('Time (hours )')

ylabel('Gene index ')

zlabel('Expression level ')

xlim([0 25]),ylim([0 700]),zlim([-5 5])

**PRACTICE**

**1.** Load the yeast microarray data described in this chapter:

a. Use traditional PCA to carry out the cluster analysis described in this chapter.

b. Using SVD identify which genes belong to each cluster of transcriptional responses (use the tools we used to cluster frames of a molecular dynamics trajectory).

c. Produce reduced data sets containing the transcriptional responses of the genes in each cluster.

**2.** Save the results from the SVD run. Repeat the analysis, this time using ICA:

a. Identify the genes corresponding to the top 20% expression levels in both types of analysis using only the 1st left singular vector and the IC vector with the best Srank.

b. Find which genes are in both sets of top expression levels (Hint: use the *'find'* and *'intersect'* functions or work with logical arrays).