Contents

Editor: Editorial de la Universidad de Granada
Autor: Kurt Stadlthanner
D.L.: Gr. 1751- 2006
ISBN: 978-84-338-4083-7

Introducción

En el campo de la genética se observa en los últimos años un enorme aumento de los métodos experimentales de alto rendimiento. A pesar de que estos métodos son muy atractivos porque permiten la medición de muchos factores o parámetros biológicos en un sólo experimento, también enfrentan a los científicos con una abrumadora cantidad de datos que puede ser difícil de analizar. Por lo tanto, se necesita desarrollar herramientas sofisticadas de análisis de datos que ayuden a los científicos a extraer automáticamente información significativa de los conjuntos de datos totales. En esta tesis se presentan nuevos enfoques para el análisis autómata de los datos de micromatrices de genes de alto rendimiento. Las micromatrices son lo último en tecnología que se usa para investigar procesos celulares a nivel genético. En primer lugar, el objetivo de un experimento con una micromatriz es determinar que genes se expresan de los miles que se encuentran en el DNA de un cierto tipo de célula. De esta manera, se usan chips modernos de micromatriz de alto rendimiento que permiten la detección cuantitativa de decenas de millar de estos genes en paralelo.

Actualmente, muchos proyectos de investigación no sólo pretenden detectar la actividad de genes individuales sino que también intentan agrupar genes

según su función biológica. Para este propósito, se divide un enorme grupo de células idénticas en varios subgrupos y se expone a cada uno de los subgrupos a un estímulo específico. Las células se adaptan a estos estímulos aumentando o disminuyendo la regulación de algunos de sus procesos biológicos, como por ejemplo, su metabolismo o algún mecanismo de reparación. Si por cada uno de los distintos estímulos se lleva a cabo un experimento con una micromatriz, los genes que pertenecen al mismo proceso biológico deberían aumentar o reducir conjuntamente y por lo tanto, tendría que ser fácil detectar mediante por ej: k-means o métodos de agrupamiento jerárquico. De todos modos, estas técnicas de agrupamiento clásicas agrupan a los genes en grupos separados, cuando se sabe que muchos genes participan en paralelo en varios procesos celulares. Por lo tanto, se requieren nuevas técnicas de agrupamiento que permitan que genes individuales aparezcan en varios grupos. Se puede resolver este problema con una agrupación basada en la separación ciega de señales (BSS,*Blind Source Separation*). Hablando en general, en BSS se asume que algunas observaciones multivariables dadas, son la suma ponderada de algunos procesos de fuentes subyacentes. Más formalmente, el modelo BSS se describe con la siguiente ecuación de matriz

$$\underline{\mathbf{X}} = \underline{\mathbf{A}}\ \underline{\mathbf{S}} \tag{1}$$

en tanto que las filas de la matriz $\underline{\mathbf{X}}$ contienen las señales observadas, las filas de la matriz $\underline{\mathbf{S}}$ contienen las señales de las fuentes subyacentes y los elementos de la matriz $\underline{\mathbf{A}}$ representan el peso con el que las fuentes individuales contribuyen a las observaciones individuales. El objetivo de una BSS es pues

recuperar las matrices desconocidas $\underline{\mathbf{A}}$ y $\underline{\mathbf{S}}$ determinando sólo la matriz de observación $\underline{\mathbf{X}}$.

Este enfoque puede aplicarse a los experimentos con micromatrices. De esta manera, los datos recopilados de una micromatriz constituyen las filas de la matriz de datos $\underline{\mathbf{X}}$, esto es, el elemento x_{mt}, $m = 1, \ldots M$, y $t = 1, \ldots, T$, contiene el nivel de expresión del gen t en la micromatriz registrada en la condición experimental m. Aquí, M indica el número de experimentos y T es el número total de genes detectados por el chip de micromatriz. Se considera entonces que las filas de la matriz $\underline{\mathbf{S}}$ contienen las huellas dactilares genéticas de los procesos biológicos individuales que ocurren en la célula. Ello significa que el elemento s_{mt}, $m = 1, \ldots, M$, $t = 1, \ldots, T$, contiene el nivel de expresión del gén t, que podría observarse si la expresión génica del proceso biológico subyacente m pudiese registrarse de forma aislada en un experimento con una micromatriz. Tal como se menciona arriba, estos procesos celulares aumentan o disminuyen dependiendo de los estímulos individuales. Estos distintos niveles de actividad están codificados en la matriz $\underline{\mathbf{A}}$, mientras que el elemento a_{ij} refleja la actividad del proceso biológico j-th bajo la condición experimental i-th. Si sólo se especifica la matriz de observación $\underline{\mathbf{X}}$, la BSS puede conducir a la determinación de las huellas dactilares genéticas desconocidas de los procesos celulares individuales (por ej.: las filas de $\underline{\mathbf{S}}$) así como su actividad bajo los distintos estímulos (elementos de $\underline{\mathbf{A}}$).

Evidentemente, el problema de BSS indicado en (1) está muy poco resuelto de modo que se requieren más suposiciones sobre las matrices $\underline{\mathbf{A}}$ y $\underline{\mathbf{S}}$. Estas suposiciones adicionales deben hacerse con cuidado ya que tienen que estar de acuerdo con los procesos biológicos reales que ocurren en una célula.

Por esta razón, en esta tesis se proporciona una minuciosa revisión de los principios de la expresión génica. Basándose en esta revisión, se estudiará la idoneidad de algunos algoritmos BSS de última tecnología para el análisis de datos de micromatriz. Como resulta estos algoritmos están basados en suposiciones que son pocos adecuados para el análisis de datos de micromatrices y que conducen a resultados difíciles de interpretar en un contexto genetico.

Por eso, se presenta un nuevo algoritmo llamado factorización dispersa de matrices no negativas (sNMF) en el que las suposiciones hechas en $\underline{\mathbf{A}}$ y $\underline{\mathbf{S}}$ se adaptan para permitir importantes aspectos de la expresión génica y regulación génica. Tal y como se mostrará en varias simulaciones, estas suposiciones son suficientes para resolver el problema de la BSS expuesto arriba. El algoritmo propuesto, así como el popular algoritmo fastICA, se aplicarán a un conjunto de datos de micromatriz del mundo real. De esta manera, se observa que sNMF conduce a resultados con más significado que el bien establecido fastICA.

Chapter 1

Bases iniciales

A menudo, cuando se analizan datos experimentales reales, transformaciones matemáticas de las observaciones ayudan a encontrar principios subyacentes. En el caso de datos univariantes muchos de estos métodos, como las transformaciones de Fourier o wavelet, filtros son conocidos y usados de manera rutinaria en el procesamiento de señales. En escenarios donde existen diferentes observaciones de un mismo proceso, es decir en casos multivariantes, la factorización de matrices se ha venido usando con mucho éxito durante los últimos años. En este tipo de factorización, la matriz de observaciones $\underline{\mathbf{X}}$ es factorizada en la matriz de mezclas $\underline{\mathbf{A}}$ y la matriz de señales subyacentes $\underline{\mathbf{S}}$, obteniendo matemáticamente:

$$\underline{\mathbf{X}} = \underline{\mathbf{A}}\ \underline{\mathbf{S}} \tag{1.1}$$

donde cada fila de la matriz $\underline{\mathbf{X}}$ $(m \times n)$, contiene una señal registrada y cada columna corresponde a una observación de la señal. Las filas de la matriz $\underline{\mathbf{S}}$

$(p \times n)$ contienen las señales transformadas, también llamadas fuentes y los elementos de la matriz $\underline{\mathbf{A}}$ $(m \times p)$ corresponden a los pesos con los cuales contribuyen las fuentes a las observaciones.

Evidentemente, el número de factorizaciones posibles de la matriz $\underline{\mathbf{X}}$ es infinito de esta forma varias restricciones adicionales son necesarias para obtener resultados de valor informativo. Por consiguiente, se han desarrollado hasta hoy una multitud de algoritmos que hacen varias susposiciones matemáticas a las matrices $\underline{\mathbf{A}}$ y $\underline{\mathbf{S}}$.

Los principios matemáticos de estos algoritmos son el tema de la primera parte del capítulo actual. Además, la base de una aplicación importante de la factorización de matrices, llamada separación ciega de señales (BSS, *blind source separation*), será descrita. En este método, las señales captadas por sensores (es decir las filas de la matriz $\underline{\mathbf{X}}$) son mezclas lineales de varias fuentes diferentes (es decir las filas de la matrix $\underline{\mathbf{S}}$). El objetivo de la separción de fuentes consiste en recuperar las señales originales y la matrix de mezcla (la matriz $\underline{\mathbf{A}}$) conociendo sólamente las observaciones. Este método se ha aplicado a muchos problemas diferentes como, por ejemplo, la separación de señales de voz de varias personas, el análisis de las grabaciones de EEG o MRT para detectar regiones activas del cerebro o la supresión de ruido e interferencias. Estas y otras aplicaciones serán expuestas al fin del capítulo.

1.1 Ámbito de la Separación Ciega de Señales

La separación ciega de señales es un problema importante en el campo del procesamiento de la señal. Por regla general, las señales captadas por sen-

sores son mezclas de varias fuentes, en principio independientes, las cuales se ven alteradas por la actuación de un medio material que las perturba. El objetivo de la separación de fuentes consiste en recuperar las señales originales partiendo de estas mezclas. Esta técnica puede ser aplicada en campos tales como el procesamiento de señales en radar, sonar y en la voz, para realzar la señal original perturbada sobre otras señales (tales como otras voces, ruidos de motores, etc...).

Inicialmente, este problema fue planteado en 1985 por los profesores Jutten, Hérault y Ans [42], como aplicación a la neurofisiología. Empleando un modelo simplificado del movimiento en la contracción de un músculo, se cuantifica dicha contracción mediante dos sensores (mezclas). A partir de las señales obtenidas, los autores tratan de obtener las fuentes, determinadas como la posición angular y la velocidad de movimiento de una articulación.

En la actualidad, el área de la Separación Ciega de Señales es aplicable a multitud de aplicaciones reales, especialmente en el campo de la ingeniería biomédica, reconocimiento y mejora de voz, econometría, sismología, minería de datos (*data mining*), etc. Las técnicas basadas en separación ciega de señales no necesitan un conjunto de datos de entrenamiento y no asumen ningún tipo de conocimiento *a priori*. Uno de los factores que hacen a esta técnica más interesante es la continúa concordancia, necesaria entre las ideas y conceptos más heurísticos con las rigurosas definiciones y propiedades matemáticas.

Lógicamente, los términos "análisis de componentes independientes" y "separación ciega de señales" no aparecieron inmediatamente. Como se mencionó anteriormente, la investigación en esta área comenzó en Francia, a me-

diados de la década de los 80, principalmente por J. Hérault, C. Jutten y B. Ans [42] pero el término "ICA" o "análisis de componentes independientes" no fue definido formalmente hasta 1990 por P. Comon [25] y ya en la comunidad internacional en 1994 [28].

1.1.1 La Independencia Estadística como Principio de Separación

En un principio puede parecer que la separación ciega de señales, o sea, la recuperación de las señales originales partiendo únicamente de un conjunto de mezclas, sin ningún conocimiento a priori pueda lograrse ya que, a primera vista, la información disponible aparenta ser insuficiente. Sin embargo, como veremos más adelante, la separación no es tan "ciega" como se intenta inducir con el propio nombre del problema, puesto que es necesario establecer una serie de hipótesis para conseguir que las salidas de nuestro sistema tengan alguna relación con las señales originales que estamos buscando.

La hipótesis esencial que domina los algoritmos de separación ciega de señales es la independencia estadística de las fuentes: los valores de las señales originales s_i no dan ninguna información acerca de los valores del resto de fuentes. Esta exigencia es considerablemente restrictiva desde el punto de vista estadístico, sin embargo no así en la práctica, ya que en la mayor parte de las situaciones es bastante factible suponer que señales generadas por distintos procesos serán independientes entre sí.

De forma sorprendentemente rápida, ya estamos en disposición de dar una solución al problema de la separación ciega de señales considerando sim-

plemente la independencia estadística de las señales originales y que no más de una de éstas tenga una función de distribución gaussiana: buscaremos un conjunto de coeficientes que transformen las señales mezcladas u observadas $\mathbf{x}$ en un conjunto de estimaciones $\mathbf{y}$ tales que dichas estimaciones sean independientes. Aunque bajo ciertas indeterminaciones (permutaciones y escalados invertibles), podemos declarar que dichas estimaciones son equivalentes a las señales originales o fuentes $\mathbf{s}$.

En cualquier caso, debe quedar claro que el concepto de independencia estadística es mucho más restrictivo que el de decorrelación. En efecto, existen diversas técnicas, como el análisis de componentes principales (PCA, *Principal Component Analysis*) o el análisis de factores basados en estadísticos de segundo orden que pueden decorrelacionar un conjunto de señales, expresadas como vectores de coeficientes reales, lo cual no implica que las señales finales sean independientes. Existe un importante caso en que decorrelación e independencia son equivalentes, y es en la situación en que las variables tienen distribuciones conjuntas gaussianas. De hecho, uno de los casos en que la técnica del análisis de componentes independientes es inservible corresponde a la separación de señales gaussianas.

Volviendo al problema de la separación ciega de señales y partiendo de la hipótesis de independencia estadística de las señales originales (fuentes), podríamos encontrar un gran número de modificaciones de las señales observadas que nos llevaran a soluciones incorreladas no independientes y que no separarían las fuentes.

Con el siguiente ejemplo se puede observar de manera más clara como dos variables están incorreladas pero no son independientes. Supongamos que las

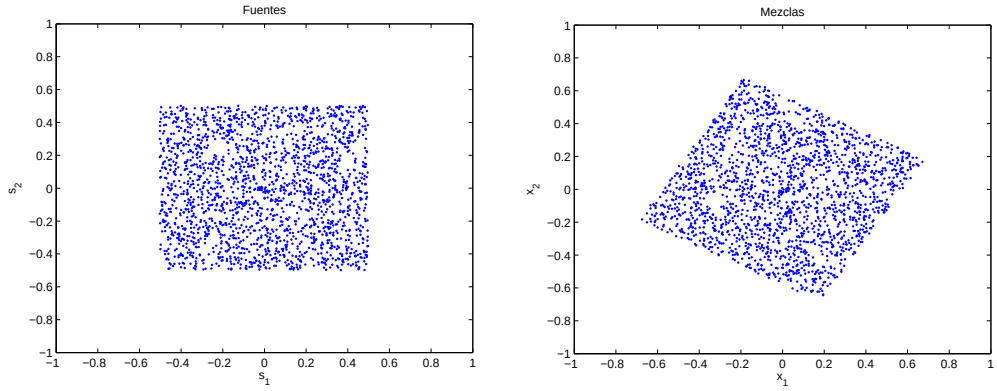

Figure 1.1: Figura izquierda: Gráfico enfrentado de dos variables de distribución uniforme independientes. Figura derecha: Mezclas tras una rotación de $\pi/3$ radians, cuyas componentes están incorreladas pero no son independientes

señales originales tienen distribuciones uniformes, tal como se muestra en la Figura 1.1. Éstas pueden ser mezcladas linealmente mediante una matriz de rotación $\underline{\mathbf{A}}$:

$$\underline{\mathbf{A}} = \begin{bmatrix} \cos(\frac{\pi}{3}) & -\sin(\frac{\pi}{3}) \\ \sin(\frac{\pi}{3}) & \cos(\frac{\pi}{3}) \end{bmatrix} \tag{1.2}$$

Las señales resultantes, están incorreladas, pero no son independientes, ya que hay un valor dado en el eje de las abscisas (por ejemplo, cercano a la esquina izquierda o derecha) el cual restringe el conjunto de valores que la componente vertical puede tomar.

1.1.2 El Análisis de Componentes Independientes y la Separación Ciega de Señales

La independencia estadística como principio básico para la resolución del problema de la separación ciega de señales relaciona estrechamente a dicho problema con el uso de la técnica de análisis de componentes independientes (*Independent Component Analysis, ICA*).

Podemos definir el análisis de componentes independientes del vector aleatorio $\mathbf{X}$ en su forma más genérica como la técnica de búsqueda de la transformación lineal $\mathbf{S} = \underline{\mathbf{W}}\mathbf{X}$ tal que las componentes S_i son lo más independientes posible, en el sentido de maximizar una determinada función $F_{\mathbf{S}}(s_1, s_2, \ldots, s_m)$ que cuantifica la independencia [46].

Se debe incidir en la generalidad de esta definición, puesto que no se establecen las hipótesis sobre el conjunto de datos y sobre el medio de mezcla.

El uso extensivo de la técnica de análisis de componentes independientes para el problema de la separación ciega de señales, hace que dichos conceptos a veces se confundan. No obstante, se debe hacer énfasis en que el análisis de componentes independientes tan sólo es la herramienta o técnica que permite resolver un problema (el de la separación ciega de señales) en el caso de asumir señales originales estadísticamente independientes. De hecho, existen numerosas líneas de investigación que van más allá de esta aparente equivalencia entre el análisis de componentes independientes y la separación ciega de señales, como es el caso que estudiaremos en este trabajo de investigación de mezclas no negativas y otros enfoques como la separación de señales dependientes.

1.1.3 Hipótesis para el Análisis de Componentes Independientes

En el análisis de componentes independientes, es necesario establecer ciertas hipótesis sobre los datos y el modo de mezcla, de forma que puedan estimar de manera adecuada las componentes independientes y que éstas coincidan con las fuentes originales desconocidas.

1. Las componentes independientes se asumen estadísticamente independientes. Esta es la hipótesis fundamental sobre la que se establece la técnica del análisis de componentes independientes. De manera intuitiva, queda claro que un conjunto de variables aleatorias $Y_1, Y_2, \ldots, Y_n$ se dice que son independientes si la información que proporciona cualquiera de las variables Y_i no proporciona a su vez ninguna información acerca de los valores de Y_j para $i \neq j$. Podemos formular matemáticamente la independencia estadística a partir de las densidades de probabilidad. Denotemos mediante $p_{\mathbf{Y}}(y_1, y_2, \ldots, y_n)$ la función de densidad de probabilidad conjunta de $Y_1, Y_2, \ldots, Y_n$ y $p_{Y_i}(y_i)$ la función de densidad de probabilidad marginal de Y_i. Decimos que los Y_i son independientes si y sólo si la función de densidad de probabilidad conjunta se puede factorizar de la siguiente manera:

$$p_{\mathbf{Y}}(y_1, y_2, \ldots, y_n) = p_{Y_1}(y_1)p_{Y_2}(y_2)\cdots p_{Y_n}(y_n) \tag{1.3}$$

2. Las componentes independientes no tienen distribuciones gaussianas. Si las componentes originales siguen distribuciones gaussianas, las mez-

clas generadas se pueden separar simplemente mediante métodos de decorrelación. Sin embargo, el análisis de componentes independientes emplea estadísticos de alto orden para los cuáles una variable que sigue una distribución normal toma valores nulos, por lo que no podría ser aplicada la técnica a este tipo de distribuciones.

3. Asumimos por simplicidad que la matriz de mezcla es cuadrada, es decir, que el número de observaciones o mezclas es igual al de componentes independientes a estimar. Está hipótesis, puede ser eliminada por otras variantes como el análisis de componentes independientes sobrecompleto (*overcomplete ICA*) en el que el número de observaciones es menor que el de componentes independientes [60],[96].

4. La matriz de mezcla es invertible. Si este no fuera el caso, alguna de las mezclas sería redundante y podría ser eliminada, volviendo al caso de un número diferente de mezclas y fuentes, que fue descartado por la tercera hipótesis.

Mediante este conjunto de restricciones podemos decir que el modelo ICA es identificable [46], esto es, que se puede estimar la matriz de mezcla y las componentes independientes salvo algunas indeterminaciones de escalado y permutaciones. Estas indeterminaciones son:

1. Permutación: el orden de las componentes independientes no se puede restabecer. Obviamente, se puede permutar el orden de las componentes independientes estimadas y éstas seguirán manteniendo la propiedad de independencia. Como el análisis de componentes independientes tan

sólo parte del conocimiento del conjunto de observaciones, resulta imposible obtener la estimación de las fuentes en el mismo orden que las originales.

2. Escalado: no es posible determinar la amplitud de las señales independientes. Efectivamente, si un conjunto de variables $Y_1, Y_2, \ldots, Y_n$ es independiente, esta se mantiene si multiplicamos dichas variables aisladamente por un coeficiente real. Nótese que incluso los escalados pueden ser invertibles, con lo que damos por válidas soluciones en las que no sólo cambia la magnitud de las señales estimadas respecto a las originales, sino también su signo. No obstante, esta indeterminación no suele ser un problema mayor en el ámbito de la separación ciega de señales, puesto que el objetivo real de la separación es recuperar la forma de las señales originales, sin tener demasiada importancia sus respectivas magnitudes. En cualquier caso, se suelen fijar las magnitudes de las estimaciones, usualmente normalizando las señales para que posean varianza unidad: $E\{Y_i^2\} = 1$.

Debido a lo anteriormente mencionado, con frecuencia las señales recuperadas aparezcan en orden distinto a las señales originales y afectadas por un factor de escala que las aumenta o disminuye. Esto es un hecho aceptado por la todos los autores que trabajan en separación ciega de fuentes [28][81], y se considera válida la separación aún cuando concurran estas dos indeterminaciones.

1.1.4 Análisis de Componentes Independientes no negativos

En muchas aplicaciones como la tomografía computerizada y el procesamiento de imagines biológicas se suponen restricciones no negativas sobre los elementos de la matriz de mezcla ($a_{ij} \geq 0$) y sobre las fuentes ($s_j \geq 0$) [21],[71],[39]. Además, varios autores han reportado recientemente que, especialmente en el campo de la análisis de imagines, el procesamiento de datos hiperespectral, el modelamiento de procesos biologicos y la codificación dispersa la factorización de observaciones $\underline{\mathbf{X}} = \underline{\mathbf{A}}\ \underline{\mathbf{S}}$ en matrices no negativas resulta en representaciones significativas y fáciles de interpretar [71],[51],[72],[39].

Si las fuentes subyacentes son independientes una extensión sencilla del procedimiento de ICA puede ser aplicada para obtener fuentes y matrices de mezcla no negativas. Para ilustrar este método suponga que los elementos a_{ij} de la matriz de mezcla y las fuentes s_j son no negativas y que las fuentes son mutuamente independientes ($p(s_i, s_j) = p(s_i)p(s_j)\ \forall i \neq j$). En este caso, un método de ICA o BSS normal es usado primero para determinar estimaciones de la matriz de mezcla $\underline{\mathbf{A}}_-$ y de las fuentes $s_{j\_}$ donde el subíndice "_" denota que las estimaciones ya pueden tener elementos negativos. Estos elementos negativos pueden aperecer en $\underline{\mathbf{A}}_-$ y $s_{j\_}$ a causa de la indeterminación de escalado inherente en la BSS incluso si la matriz de mezcla $\underline{\mathbf{A}}$ y las fuentes s_j originales son estrictamente no negativas. De este manera, columnas enteras de $\underline{\mathbf{A}}_-$ pueden llegar a ser negativas con tal de que las fuentes corespondientes

seán negativas tambíen, p. ej.:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{bmatrix} a_{11} & -a_{12} \\ a_{21} & -a_{22} \end{bmatrix} \begin{pmatrix} s_1 \\ -s_2 \end{pmatrix} \tag{1.4}$$

Por tanto, la multiplicación de las columnas negativas de $\underline{\mathbf{A}}$ (la segunda en el ejemplo superios) y las correspondientes fuentes $s_{j\_}$ (s_2 en el mencionado ejemplo) por -1 conduce a la deseada no negatividad.

Entonces, no algoritmo específico tiene que ser desarrallado por la ICA no negativa si las fuentes subyacentes son independientes.

1.1.5 Factorización de matrices no negativas

La ICA no negativa presentada en la sección anterior solamente puede ser aplicada si las fuentes subyacentes son mutuamente independientes. No obstante, existen también algunos algoritmos que son capaz de factorizar una matriz no negativa $\underline{\mathbf{X}}$ en dos matrices no nogativas $\underline{\mathbf{A}}$ y $\underline{\mathbf{S}}$ sin ninguna restricción addicional:

$$\underline{\mathbf{X}} = \underline{\mathbf{A}} \ \ \underline{\mathbf{S}}, \qquad \underline{\mathbf{X}}, \underline{\mathbf{A}}, \underline{\mathbf{S}} \text{ nonnegative.} \tag{1.5}$$

Sin embargo, estos metodos no conducen a resultados unicos, es decir, que se pueden encontrar varias matrizes $\underline{\mathbf{A}}$ y $\underline{\mathbf{S}}$ no negativas que factorizan $\underline{\mathbf{X}}$ perfectamente. No obstante, estos algoritmos, también llamados algoritmos de la factorización de matrices no negativas (NMF, *Nonnegativ Matrix Factorisation*), se han usado con exito en el campo de la espectrosopía o en el

campo del procesamiento de imágines de caras [40][52][51]. A menudo, las matrices $\underline{\mathbf{S}}$ y $\underline{\mathbf{A}}$ obtenido por la NMF son dispersas (es decir que la mayor parte de sus elementos valen cero) y por eso son fáciles de interpretar.

1.2 Principios Matemáticos

En esta sección se presentan los principios matemáticos necesarios para entender los algoritmos de factorización de matrices presentados más adelante. De este modo, algunos de las propiedades basicas de las variables aleatorias serán introducidas y las medidas de distancia y de dispersión serán explicados.

1.2.1 Definición de Variables Aleatorias

Una variable es aleatoria si toma diferentes valores como resultado de un experimento aleatorio. Matemáticamente, es una aplicación

$$X : \Omega \rightarrow \mathbb{R} \tag{1.6}$$

que da un valor numérico, del conjunto de los reales $\mathbb{R}$, a cada suceso en el espacio Ω de los resultados posibles del experimento. Las variables aleatorias son usualmente designadas por letras mayúsculas (e.g. X, Y, ...), y un valor particular de una variable aleatoria es denominada por una letra minúscula (e.g. x, y, ...).

1.2.2 Función de Distribución Acumulada

La función de distribución acumulada (cdf, *cumulative distribution function*) $F_X(x_0)$ corresponde a la probabilidad de que una variable aleatoria real X tome un valor numérico menor o igual a x_0, o representa la suma de las probabilidades hasta alcanzar el valor de interés. Simbólicamente, lo anterior se expresa como:

$$F_X(x_0) = P(X \leq x_0) \text{ para } x_0 \in \mathbb{R}. \tag{1.7}$$

La función de probabilidad acumulada $F_X(x)$ cumple con las siguientes propiedades:

1. $F_X(x_0)$ es no negativo:

$$F_X(x_o) \geq 0 \quad \forall\ x_0 \in \mathbb{R}.$$

2. F_X es no decreciente:

$$F_X(x_1) \leq F_X(x_2) \quad \forall\ x_1 \leq x_2, \quad x_1, x_2 \in \mathbb{R}.$$

3. El recorrido de la función de probabilidad acumulada es

$$0 \leq F_X(x_0) \leq 1 \quad \forall\ x_0 \in \mathbb{R}.$$

4. El valor de la función de probabilidad acumulada cuando el valor de la

variable es demasiado grande se acerca a uno:

$$F_X(\infty) = 1.$$

5. El valor de la función de probabilidad acumulada cuando el valor de la variable es demasiado pequeño se acerca a cero:

$$F_X(-\infty) = 0.$$

1.2.3 Función de Densidad de Probabilidad

La función de densidad de probabilidad (fdp, pdf, *probability density function*) p_X de una variable aleatoria continua X es una función que se integra para obtener la probabilidad que la variable aleatoria toma un valor en un intervalo predefinido:

$$\int_{x_1}^{x_2} p_X(t)dt = P(x_1 \leq X \leq x_2). \tag{1.8}$$

La fdp está relacionada con la función de distribución de probabilidad como sigue:

$$p_X(x_0) = \left.\frac{\mathrm{d}F_X(x)}{\mathrm{d}x}\right|_{x=x_0}. \tag{1.9}$$

De manera análoga, se puede relacionar de forma inversa ambas funciones:

$$F_X(x_0) = \int_{-\infty}^{x_0} p_X(t)dt. \tag{1.10}$$

La fdp cumple con las siguientes propiedades:

1. p_X es no negativo

$$p_X(x_0) \geq 0 \quad \forall\ x_0 \in \mathbb{R}.$$

2. El recorrido de la fdp es

$$0 \leq p_X(x_0) \leq 1 \quad \forall\ x_0 \in \mathbb{R}.$$

3. La fdp está normalizada

$$\int_{-\infty}^{\infty} p_X(t)dt = 1.$$

1.2.4 Vectores Aleatorios

En la práctica, la naturaleza aleatoria de un proceso puede necesitar ser descrito por más de una variable aleatoria. Por eso, se define un vector aleatorio como sigue: Sean $X_1, \ldots, X_n$ variables aleatorias definidos sobre el mismo espacio probabilístico. El vector ordenado $[X_1, \ldots, X_N]$ se llama variable aleatoria N-dimensional o vector aleatorio de dimensión N.

Más formalmente, un vector aleatorio es una función con domino del espacio probabilístico Ω y destino $\mathbb{R}^N$:

$$[X_1, \ldots, X_N] : \Omega \to \mathbb{R}^N. \tag{1.11}$$

1.2.5 Funciones de Distribución y Densidad de Vectores Aleatorios

En el caso multivariable se define la función de distribución acumulada y la función de densidad de probabilidad en analogía con el caso univariable:

Sea $\mathbf{X} = [X_1, \ldots, X_N]$ un vector aleatorio real y sean $X_1, \ldots, X_N$ variables reales aleatorias de un mismo experimento estadístico. La función de distribución acumulada de $\mathbf{X}$ se define como

$$F_{\mathbf{X}}(x_1, \ldots, x_N) = P(X_1 \leq x_1, X_2 \leq x_2, \ldots, X_N \leq x_N). \tag{1.12}$$

De acuerdo con esta definición la función de probabilidad de un vector aleatorio puede ser determinado como en la ecuación (1.9) si se sustituyen las derivadas totales por derivadas parciales respecto a cada una de las variables:

$$p_{\mathbf{X}}(x_{1_0}, x_{2_0}, \ldots, x_{N_0}) = \left.\frac{\partial F_{\mathbf{X}}}{\partial x_1}\right|_{x_1=x_{1_0}} \left.\frac{\partial F_{\mathbf{X}}}{\partial x_2}\right|_{x_2=x_{2_0}} \ldots \left.\frac{\partial F_{\mathbf{X}}}{\partial x_N}\right|_{x_N=x_{N_0}}. \tag{1.13}$$

1.2.6 Funciones de Distribución y Densidad Conjuntas y Marginales

Los conceptos anteriores pueden ser extendidos también a problemas con dos vectores aleatorios diferentes $\mathbf{X}$ y $\mathbf{Y}$ donde la dimensión m del vector $\mathbf{Y}$ puede ser diferente que la dimensión n del vector $\mathbf{X}$. Si se concatenan los dos vectores para obtener un "supervector" $\mathbf{Z} = [\mathbf{X}, \mathbf{Y}]$ las fórmulas precedentes pueden ser utilizadas directamente.

En tal escenario se definen la función de distribución conjunta

$$F_{\mathbf{X},\mathbf{Y}}(\mathbf{x}_0, \mathbf{y}_0) = P(\mathbf{X} \leq \mathbf{x}_0, \mathbf{Y} \leq \mathbf{y}_0) \tag{1.14}$$

y la función de densidad conjunta

$$p_{\mathbf{X},\mathbf{Y}}(\mathbf{x}_0, \mathbf{y}_0) = \frac{\partial F_{\mathbf{X},\mathbf{Y}}(\mathbf{x}, \mathbf{y})}{\partial \mathbf{x}}\bigg|_{\mathbf{x}=\mathbf{x}_0} \frac{\partial F_{\mathbf{X},\mathbf{Y}}(\mathbf{x}, \mathbf{y})}{\partial \mathbf{y}}\bigg|_{\mathbf{y}=\mathbf{y}_0} \tag{1.15}$$

donde $\mathbf{x}_0$ e $\mathbf{y}_0$ son un vector constante de valores concretos del conjunto de variables que forman $\mathbf{X}$ e $\mathbf{Y}$, respectivamente. Asimismo, la función de distribución se deriva de la función de densidad conjunta

$$F_{\mathbf{X},\mathbf{Y}}(\mathbf{x}_0, \mathbf{y}_0) = \int_{-\infty}^{\mathbf{x}_0} \int_{-\infty}^{\mathbf{y}_0} p_{\mathbf{X},\mathbf{Y}}(\boldsymbol{\mu}\boldsymbol{\nu}) d\boldsymbol{\mu} d\boldsymbol{\nu}. \tag{1.16}$$

Dado la función de densidad conjunta las dos funciónes de densidad de probabilidad individuales $p_{\mathbf{X}}(\mathbf{x})$ de $\mathbf{X}$ y $p_{\mathbf{Y}}(\mathbf{y})$ de $\mathbf{Y}$, llamadas funciónes de densidad marginales, pueden ser también obtenidas. Por eso, se integra la función de densidad conjunta sobre uno de los dos vectores aleatorios como sigue

$$p_{\mathbf{X}}(\mathbf{x}) = \int_{-\infty}^{\infty} p_{\mathbf{X},\mathbf{Y}}(\mathbf{x}, \boldsymbol{\mu}) d\boldsymbol{\mu} \tag{1.17}$$

$$p_{\mathbf{Y}}(\mathbf{y}) = \int_{-\infty}^{\infty} p_{\mathbf{X},\mathbf{Y}}(\boldsymbol{\nu}, \mathbf{y}) d\boldsymbol{\mu} \tag{1.18}$$

1.2.7 Esperanzas

En esta sección serán expuestas las esperanzas de variables y vectores aleatorios. Estas esperanzas se utilizan frecuentemente en la práctica ya que pueden

ser estimadas de observaciones de variables aleatorias incluso si la fdp de dichas variables no se conoce de manera exacta.

Generalmente, la esperanza de una magnitud $\mathbf{g}(\mathbf{x})$ derivada del vector aleatorio $\mathbf{X}$ se define como:

$$E\{\mathbf{g}(\mathbf{x})\} = \int_{-\infty}^{\infty} \mathbf{g}(\mathbf{x}) p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}. \tag{1.19}$$

Aquí la magnitud $\mathbf{g}(\mathbf{x})$ puede ser una cantidad escalar, un vector o una matriz.

Un caso especial de las esperanzas son los momentos de un vector aleatorio. Estos momentos se obtienen si $\mathbf{g}(\mathbf{x})$ es el resultado del producto de componentes del vector aleatorio $\mathbf{X}$. Generalmente, el momento de orden n de un vector aleatorio definido como

$$\mathbf{m}_{\mathbf{X}}^{\{n\}} = \int_{-\infty}^{\infty} \mathbf{x}^n p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}. \tag{1.20}$$

En detalle, el primer momento $\boldsymbol{\mu}_{\mathbf{X}} := \mathbf{m}_{\mathbf{X}}^{\{1\}}$ corresponde al vector media del vector $\mathbf{X}$ mientras que el segundo momento define la correlación $\underline{\mathbf{R}}_{\mathbf{X}}$ entre pares de componentes del vector aleatorio $\mathbf{X} = [X_1, \ldots, X_N]^T$:

$$\underline{\mathbf{R}}_{\mathbf{X}} = (r_{ij})_{N \times N} = \mathbf{m}_{\mathbf{X}}^{\{2\}} = E\{\mathbf{X}\mathbf{X}^T\} \tag{1.21}$$

donde

$$r_{ij} = E\{X_i X_j\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_i x_j p_{X_i, X_j}(x_i, x_j) dx_j dx_i. \tag{1.22}$$

Esta matriz $\underline{\mathbf{R}}_{\mathbf{X}}$ tiene algunas propiedades importantes:

1. $\underline{\mathbf{R}}_{\mathbf{X}}$ es simétrica.

2. $\underline{\mathbf{R}}_{\mathbf{X}}$ es semidefinida positiva.

3. Todos los valores propios de $\underline{\mathbf{R}}_{\mathbf{X}}$ son reales y no negativos.

4. Todos los vectores propios de $\underline{\mathbf{R}}_{\mathbf{X}}$ son reales y ortogonales.

A menudo, la media está restando a los vectores aleatorios antes que los momentos están determinados. Estos momentos se nombran momentos centrados y son solamente significativos si sus ordenes son más grandes que 1. Correspondiendo al momento de segundo orden, correlación, se denomina al momento centrado de segundo orden covarianza. La correspondiente matriz de covarianza $\underline{\mathbf{C}}_{\mathbf{X}}$ se calcula como sigue:

$$\underline{\mathbf{C}}_{\mathbf{X}} = (c_{ij})_{N\times N} = E\{(\mathbf{X} - \boldsymbol{\mu}_{\mathbf{X}})(\mathbf{X} - \boldsymbol{\mu}_{\mathbf{X}})^T\} \tag{1.23}$$

donde los elementos

$$c_{ij} = E\{(X_i - \mu_i)(X_j - \mu_j)\} \tag{1.24}$$

se llaman covarianzas. Esta matriz $\underline{\mathbf{C}}_{\mathbf{X}}$ tiene las mismas propiedades que la matriz $\underline{\mathbf{R}}_{\mathbf{X}}$.

Nótese que en el caso especial de una sola variable aleatoria X el vector media se reduce al valor medio $\mu_X = E\{X\}$, la matriz de correlación se reduce al momento segundo $E\{X^2\}$ y la matriz de covarianza se reduce a la variación de X

$$\sigma_X^2 = E\{(X - \mu_X)^2\}. \tag{1.25}$$

Además, la raíz σ_X de la variación se denomina desviación estándar.

1.2.8 Distribuciones de Probabilidad Importantes

En esta sección se exponen algunas de las funciones de densidad de probabilidad más importantes. Estas funciones se utilizan con mucha frecuencia en el procesamiento de señales.

1.2.8.1 Distribución Uniforme

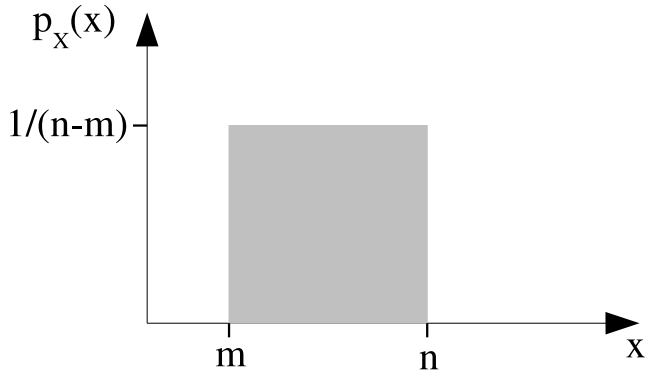

Figure 1.2: Función de densidad de probabilidad uniforme.

Las distribuciones uniformes corresponden al experimento de elegir dos puntos al azar entre dos fijos m y n. Como la probabilidad de elegir cualquier punto es la misma, la función de densidad tendrá la misma altura en todos los puntos entre m y n, es decir se trata de una función constante desde m a n, de altura $1/(m-n)$ (ver Figura 1.2):

$$p_X(x) = \begin{cases} \frac{1}{m-n} & \text{si} \quad x \in [m,n] \\ 0 & \text{resto} \end{cases} \tag{1.26}$$

Aquí, la media coincide con el punto medio del segmento $[m, n]$

$$\mu_X = \frac{n-m}{2}, \tag{1.27}$$

y la varianza se determine como

$$\sigma_X^2 = \frac{(n-m)^2}{12}. \tag{1.28}$$

1.2.8.2 Distribución Normal

La distribución normal multivariable, también llamada distribución de Gauss o distribución gaussiana, tiene algunas propiedades específicas que la hacen única entre todas las funciones de densidad de probabilidad. Por eso, será expuesta en detalle en esta sección.

Sea $\mathbf{X}$ un vector aleatorio de dimensión n. $\mathbf{X}$ se llama vector aleatorio gaussiano si su función de probabilidad de densidad está determinada por la siguiente expressión:

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}(\det \underline{\mathbf{C}}_{\mathbf{X}})^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_{\mathbf{X}})^T \underline{\mathbf{C}}_{\mathbf{X}}^{-1}(\mathbf{x}-\boldsymbol{\mu}_{\mathbf{X}})\right), \tag{1.29}$$

donde $\det \underline{\mathbf{C}}_{\mathbf{X}}$ es la determinante de $\underline{\mathbf{C}}_{\mathbf{X}}$. Nótese, que la matriz de covarianza $\underline{\mathbf{C}}_{\mathbf{X}}$ se supone definida positiva implicando que su inversa existe.

Por un vector gaussiano $\mathbf{X}$ puede ser demostrado que

$$E\{\mathbf{X}\} = \boldsymbol{\mu}_{\mathbf{X}}, \qquad E\{(\mathbf{X}-\boldsymbol{\mu}_{\mathbf{X}})(\mathbf{X}-\boldsymbol{\mu}_{\mathbf{X}})^T\} = \underline{\mathbf{C}}_{\mathbf{X}} \tag{1.30}$$

Por eso, $\boldsymbol{\mu}_{\mathbf{X}}$ y $\underline{\mathbf{C}}_{\mathbf{X}}$ se llaman vector media y matriz de covarianza respecti-

vamente, de la distribución normal multivariada. Además, el conocimiento de $\boldsymbol{\mu}_{\mathbf{X}}$ y $\underline{\mathbf{C}}_{\mathbf{X}}$ ya es suficiente para definir completamente la fdp gaussiana, es decir, que momentos de orden superior no proporcionan ninguna información sobre dicha fdp.

La razón más importante para la popularidad de la fdp gaussiana proviene del teorema del límite central que indica que la distribución de la suma de variables aleatorias tiende a una distribución gaussiana. Más formalmente, sea X_k la suma de la secuencia de variables aleatorias idénticamente distribuidas $\{Z_i\}$:

$$X_k = \sum_{i=1}^{k} Z_i. \tag{1.31}$$

Como la media μ_X y la varianza σ_X de X_k puede crecer sin límite en el caso que $k \to \infty$, se considera la variable normalizada Y_k en lugar de X_k:

$$Y_k = \frac{X_k - \mu_{X_k}}{\sigma_{X_k}}. \tag{1.32}$$

En este caso el teorema del límite central afirma que la distribucíon de Y_k converge a una distribución normal con media cero y varianza unidad cuando $k \to \infty$.

Este teorema se extiende fácilmente a vectores aleatorios idénticamente distribuidos $\mathbf{Z}_k$ con media $\boldsymbol{\mu}_{\mathbf{Z}}$ y matriz de variance $\underline{\mathbf{C}}_{\mathbf{Z}}$ común. En este caso, la distribución del vector aleatorio

$$\mathbf{Y}_k = \frac{1}{\sqrt{k}} \sum_{i=1}^{k} (\mathbf{Z}_i - \boldsymbol{\mu}_Z) \tag{1.33}$$

converge a una distribución gaussiana con media cero y matriz de covarianza

$\underline{\mathbf{C}}_{\mathbf{Z}}$.

Este teorema es la razón de que se asuma que muchas de las variables observadas en el mundo real siguan una distriución gaussiana. Al ser la suma de un conjunto suficiente de otras variables desconocidas, sin importar sus respectivas funciones de densidad de probabilidad, se puede asegurar que la variable que corresponde a la suma sigue una distribución similar a la normal.

Las consecuencias del teorema del límite central en BSS son profundas, puesto que se puede asumir que, con un número suficiente de señales participando en la mezcla, ésta tenderá a ser gaussiana. Por eso, un primer método intuitivo para recuperar las señales que dieron origen a la mezcla es tratar de alejar las distribuciones de las estimaciones lo más posible de la distribucíon gaussiana, es decir, aumentar la no-guassianidad.

1.2.8.3 Distribución Laplaciana

Otra distribución importante en el campo del procesamiento de datos es la distribución laplaciana:

$$p_X(x) = \frac{\lambda}{2} \exp(-\lambda|x|) \tag{1.34}$$

Esta distribución con su grande concentración de valores en el centro y sus alrededores (suponemos que la media de la señal en cuestión es precisamente cero) se usa a menudo para describir los señales de voz.

1.2.9 Correlaciones y Covarianzas Cruzadas

Basandose en la función de densidad conjunta (ver ecuación (1.15)) las operaciones de esperanzas pueden también ser extendidos a funciones $\mathbf{g}(\mathbf{x}, \mathbf{y})$ de dos vectores aleatorios $\mathbf{X}$ y $\mathbf{Y}$ como sigue:

$$E\{\mathbf{g}(\mathbf{x}, \mathbf{y})\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{g}(\mathbf{x}, \mathbf{y}) p_{\mathbf{X},\mathbf{Y}}(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}. \tag{1.35}$$

Análogamente, se definen la matriz de correlaciones cruzadas y las covarianzas cruzadas respectivamente como:

$$\underline{\mathbf{R}}_{\mathbf{X},\mathbf{Y}} = E\{\mathbf{X}\mathbf{Y}^T\}, \tag{1.36}$$

$$\underline{\mathbf{C}}_{\mathbf{X},\mathbf{Y}} = E\{(\mathbf{X} - \boldsymbol{\mu}_X)(\mathbf{Y} - \boldsymbol{\mu}_Y)^T\}. \tag{1.37}$$

La medida más habitualmente utilizada para cuantificar la correlación cruzada entre dos variables aleatorios X y Y es el coeficiente de correlación lineal de Pearson $r_{Pearson}$. El coeficiente de Pearson mide el grado de asociación lineal entre dos variables cualesquiera, y puede calcularse dividiendo la covarianza de ambas entre el producto de las desviaciones típicas de las dos variables:

$$r_{Pearson}(X, Y) = \frac{C_{X,Y}}{\sigma_X \sigma_Y}. \tag{1.38}$$

$r_{Pearson}(X, Y)$ tiene las propiedades siguientes:

1. El recorrido de $r_{Pearson}(X, Y)$ es

$$-1 \leq r_{Pearson}(X, Y) \leq 1 \quad \forall X, Y \in \mathbb{R}$$

2. $r_{Pearson}(X, Y)$ es positivo si existe una relación directa entre ambas variables y es negativo si la relación es inversa.

3. Un valor de $r_{Pearson}(X, Y)$ de +1 o -1 indica una relación lineal perfecta entre X y Y, mientras que un valor 0 indica que no existe una relación lineal.

Finalmente, nótese que un valor de cero de $r_{Pearson}$ no indica necesariamente que no exista correlación entre X y Y, ya que las variables pueden presentar una relación no lineal.

1.2.10 Estimación de Funciones Estadísticas

Las correlaciones y covarianzas se emplean como herramientas estadísticas para determinar las posibles dependencias lineales entre las distintas variables aleatorias. Hasta ahora, todas las fórmulas de momentos se han determinado basándose en el conocimiento de la función de densidad de probabilidad. Como ya se comentaba al principio del 1.2.7, normalmente no se conoce esta función. No obstante, es posible realizar una estimación de estos valores si se conoce un conjunto $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K$ de valores que toma el conjunto de variables que forman el vector aleatorio $\mathbf{X}$.

En este caso, tienen que ser distinguidos dos estimadores diferentes dependiendo de que se trate de un momento normal o con un momento centrado. En el primer caso, el de momentos normales de orden n, se usa el estimador

$$\frac{1}{K}\sum_{k=1}^{K}\mathbf{x}_i^n \tag{1.39}$$

mientras que en el caso de momentos centrados de orden n se aplicaría la función

$$\frac{1}{K-1}\sum_{k=1}^{K}(\mathbf{x}_i - \boldsymbol{\mu}_{\mathbf{x}})^n. \tag{1.40}$$

Por ejemplo, el estimador de $\boldsymbol{\mu}_{\mathbf{X}} = E\{\mathbf{X}\}$ es

$$\hat{\boldsymbol{\mu}}_{\mathbf{X}} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}_k \tag{1.41}$$

y la covarianza cruzada entre dos variables aleatorias X y Y se determina como

$$\hat{c} = \frac{1}{K-1}\sum_{k=1}^{K}(x_k - \mu_X)(y_k - \mu_Y). \tag{1.42}$$

1.2.11 Decorrelación y Blanqueado

Decimos que dos vectores $\mathbf{X}$ e $\mathbf{Y}$ están incorrelados si su matriz de covarianzas cruzadas, $\underline{\mathbf{C}}_{\mathbf{X},\mathbf{Y}}(\mathbf{x},\mathbf{y})$, es la matriz nula. De forma equivalente, podemos decir que dichos vectores $\mathbf{X}$ e $\mathbf{Y}$ están incorrelados si su matriz de correlaciones cruzadas es $\underline{\mathbf{R}}_{\mathbf{X},\mathbf{Y}} = \boldsymbol{\mu}_{\mathbf{X}}\boldsymbol{\mu}_{\mathbf{Y}}$.

Si nos referimos al concepto de decorrelación entre los componentes de un mismo vector aleatorio $\mathbf{X}$, nos referimos a la matriz de covarianzas definida en la Ecuación (1.23). Obviamente, cada uno de los componentes X_i de $\mathbf{X}$ está perfectamente correlado consigo mismo, por lo que la condición de

decorrelación relativa a la matriz de covarianza es:

$$\mathbf{X} \text{ ésta decorrelado sii } \underline{\mathbf{C}}_{\mathbf{X}=(X_1,\ldots,X_n)^T} = \begin{bmatrix} \sigma^2_{X_1} & 0 & \cdots & 0 \\ 0 & \sigma^2_{X_2} & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \sigma^2_{X_n} \end{bmatrix} \tag{1.43}$$

donde $\sigma_{X_i} = E\{X_i - \mu_{X_i}\}$ corresponde al momento central de segundo orden, o sea, a la varianza.

Como caso particular, se dice que un vector aleatorio está blanqueado si su media es cero para todos sus componentes y su matriz de covarianzas es la matriz identidad $\underline{\mathbf{I}}$. Al imponer que la media sea cero, lógicamente la matriz de correlación coincide con la de covarianza.

$$\mathbf{X} \text{ está blanqueado si } \boldsymbol{\mu}_X = 0,\ \underline{\mathbf{C}}_{\mathbf{X}} = \underline{\mathbf{R}}_{\mathbf{X}} = \underline{\mathbf{I}}. \tag{1.44}$$

El blanqueo de las señales suele ser un paso habitual de preprocesamiento de los datos en el análisis de componentes independientes, en muchos casos necesario para realizar estimaciones correctas de independencia.

1.2.12 Independencia Estadística

Decimos que un vector aleatorio $\mathbf{X}$ es independiente si la función de densidad de probabilidad conjunta de $\mathbf{X}$ es factorizable en el producto de las funciones de densidad de probabilidad marginales de sus componentes:

$$\mathbf{X} \text{ es independiente si } p_{\mathbf{X}}(\mathbf{x}) = p_{X_1}(x_1)p_{X_2}(x_2) \cdot \ldots \cdot p_{X_n}(x_n) = \prod_{i=1}^{n} p_{X_i}(x_i) \tag{1.45}$$

De igual manera, se podría expresar la propiedad de independencia si la función de distribución de probabilidad conjunta de $\mathbf{X}$ se puede factorizar en el producto de las funciones de distribución de probabilidad marginales de sus componentes, si bien es más habitual emplear la expresión (1.45).

Claramente la condición de independencia expresada en (1.45) es mucho más restrictiva que la de decorrelación (1.43). De hecho, es posible definir la independencia como la decorrelación no lineal [42], es decir, si dos variables X e Y son independientes, entonces cualesquiera transformaciones no lineales $g(x)$ y $g(y)$ están incorreladas entre sí.

Nótese que en el caso especial de que las variables tengan distribuciones gaussianas los conceptos de decorrelación e independencia coinciden.

1.2.13 Estadística de Alto Orden

Los estadísticos que se han descrito hasta ahora corresponden a momentos de primer o segundo orden (media, varianza, correlación, covarianza). Nos referimos a estadísticos de alto orden cuando se emplean momentos de orden mayor que dos. Éstos contienen información del tipo de la desviación de la distribución de una determinada variable respecto de la de una distribución gaussiana o del grado de simetría de la función de densidad de probabilidad. Normalmente, el conocimiento de un conjunto finito de momentos de una determinada variable aleatoria es, para la mayoría de tipos de distribuciones,

equivalente al conocimiento de la función de densidad de probabilidad [66].

1.2.13.1 Coeficiente de Asimetría

El coeficiente de asimetría o apuntamiento corresponde al momento central de tercer orden sobre una variable aleatoria:

$$\mu_3 = E\{(X - \mu_X)^3\} \tag{1.46}$$

El coeficiente de asimetría vale 0 para funciones de densidad de probabilidad simétricas respecto a la media, mientras que un valor negativo indica que la distribución está descompensada hacia la izquierda respecto a la media y un valor positivo indica desplazamiento a la derecha.

1.2.13.2 Kurtosis

La kurtosis se calcula en relación al momento de cuarto orden de acuerdo a la siguiente expresión:

$$kurt(X) = \mu_X - 3\sigma^4 = E\{X^4\} - 3\left[E\{X^2\}\right]^2. \tag{1.47}$$

En realidad existen diversas formulaciones para la kurtosis, según esté normalizada o no, pero lo que realmente interesa de esta magnitud es su cercanía al cero, puesto que es una medida de no-gaussianidad. De esta forma, queda patente su importancia para la separación ciega de señales y el análisis de componentes independientes, al relacionarse la no-gaussianidad con la separación efectiva de las fuentes (1.2.8.2).

Una variable aleatoria cuya kurtosis sea cero, tiene una distribución mesocúrtica, probablemente gaussiana. Si la kurtosis es negativa, la distribución es platicúrtica o subgaussiana (por ejemplo, la distribución uniforme). Por último, si la kurtosis es positiva, su respectiva distribución se dice que es leptocúrtica o supergaussiana (por ejemplo, la distribución laplaciana).

1.2.13.3 Cumulantes

Es posible obtener una función genérica para la generación de momentos de cualquier orden mediante la transformada de Fourier continua de la función de densidad de probabilidad de X. A dicha función la denominamos función generadora de momentos:

$$\phi(\omega) = E\{\exp(i\omega x)\} = \int_{-\infty}^{\infty} \exp(i\omega x) p_X(x) dx, \tag{1.48}$$

donde $i = \sqrt{-1}$ y ω corresponde a la variable x tras la transformación.

Los cumulantes corresponden a otro tipo de estadísticos de alto orden derivados a partir del logaritmo neperiano de la función generadora de momentos, de la misma forma que los momentos se pueden obtener de dicha función:

$$\Phi(\omega) = \ln(\phi(\omega)) \tag{1.49}$$

Para una variable aleatoria X de media 0, los primeros 4 cumulantes son:

$$\begin{aligned}
\kappa_1 &= 0, \\
\kappa_2 &= E\{X^2\}, \\
\kappa_3 &= E\{X^3\}, \\
\kappa_4 &= E\{X^4\} - 3\left[E\{X^2\}\right]^2.
\end{aligned} \tag{1.50}$$

Queda claro que los tres primeros cumulantes coinciden con sus respectivos momentos, mientras que el cuarto corresponde a la kurtosis definida en el apartado anterior.

En el caso multivariado, es decir, cuando tratamos un vector $\mathbf{X}$ de variables aleatorias, extrapolamos las fórmulas de los cumulantes, obteniendo los cumulantes cruzados. El segundo y tercer cumulante cruzado coincide con sus respectivos momentos cruzados. La expresión del cumulante cruzado de cuarto orden cambia respecto a su correspondiente momento cruzado:

$$\begin{aligned}
cum(X_i, X_j) &= E\{X_i X_j\}, \\
cum(X_i, X_j, X_k) &= E\{X_i X_j X_k\}, \\
cum(X_i, X_j, X_k, X_l) &= E\{X_i X_j X_k X_l\} - E\{X_i X_j\}E\{X_k X_l\} \\
&\quad - E\{X_i X_k\}E\{X_j X_l\} - E\{X_i X_l\}E\{X_j X_k\}
\end{aligned} \tag{1.51}$$

Obviamente, al poder obtener los cumulantes a partir de los momentos, ambos contienen la misma información estadística. Sin embargo, es preferible tratar con cumulantes puesto que presentan una serie de propiedades

deseables que no se encuentran en los momentos, entre ellas la de que todos los cumulantes de orden mayor que tres de un vector aleatorio $\mathbf{X}$ de distribución gaussiana son nulos.

1.2.14 Teoría de la Informacíon: Entropía e Información Mutua

La teoría de la información es la disciplina científica que estudia la información y todo lo relacionado con ella. La información es tratada como una magnitud física y el objetivo es codificar las observaciones de la manera más conveniente. Los problemas que plantea Claude E. Shannon, quien formuló esta teoría en 1940, tienen que ver con la cantidad de información, la capacidad del canal de comunicación, el proceso de codificación que puede utilizarse para cambiar el mensaje en una señal y los efectos del ruido.

1.2.14.1 Entropía

Dada una variable aleatoria discreta X que tiene una determinada distribución de probabilidades, $p_X(x)$, la entropía de X se define como:

$$H(X) = -\sum_{x \in X} p_X(x) \log_b p_X(x). \tag{1.52}$$

De esta forma, la entropía de una variable aleatoria se puede interpretar como la cantidad de información que proporciona la observación de dicha variable. Bajo otro punto de vista, también es una medida del "desorden" de una variable, puesto que cuanto más impredecible sea el valor de una instancia determinada de la variable, mayor será su entropía. La unidad en

que se mide la información depende de la base del logaritmo utilizada. Si la base es 2, entonces la unidad de medida es el BIT, o si la base es el número e, la unidad correspondiente es el NAT. No obstante, para el análisis de componentes independientes no es relevante la base que empleemos puesto que tan sólo cambia la escala de medida.

1.2.14.2 Entropía Diferencial

La definición anterior y la expresión (1.52) pueden ser generalizadas para variables aleatorias continuas y vectores de variables aleatorias continuas, en cuyo caso se suele denominar a esta medida entropía diferencial:

$$H(\mathbf{X}) = -\int p_{\mathbf{X}}(\mathbf{x}) \log_b p_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}. \tag{1.53}$$

1.2.14.3 Teorema de Disminución de la Entropía

La entropía de una variable X condicionada por otra Y es menor o igual que la entropía de X, alcanzándose la igualdad si y sólo si las variables X e Y son independientes.

$$\begin{aligned} &H(X|Y) \leq H(X) \\ &H(X|Y) = H(X) \Leftrightarrow X \text{ e } Y \text{ son independientes} \end{aligned} \tag{1.54}$$

En términos intuitivos podemos interpretar el teorema afirmando que el conocimiento de una experiencia sólo puede disminuir nuestra incertidumbre sobre otra cualquiera, siendo esta reducción efectiva siempre y cuando ambas experiencias no tengan ninguna relación.

Corolario: Como importante corolario para el entorno del análisis de componentes independientes obtenemos que la entropía de una variable aleatoria n-dimensional es menor o igual que la suma de las entropías marginales de sus componentes, alcanzándose la igualdad si y sólo si éstas son independientes:

$$H(\mathbf{X} = [X_1, X_2, \dots, X_n]) \leq \sum_{i=1}^{n} H(X_i). \tag{1.55}$$

1.2.14.4 Negentropía

Entre todas las variables aleatorias con una matriz de covarianza fija, la función de densidad de probabilidad de una variable gaussiana tiene entropía máxima. De esta orma, podemos entender la entropía como una medida de gaussianidad, siendo la distribución gaussiana la más desestructurada o más aleatoria.

Se define el concepto de negentropía a partir del de entropía para determinar aquella medida no negativa, que es cero para un vector aleatorio gaussiano, determinada por la siguiente expresión:

$$J(\mathbf{X}) = H(\mathbf{X}_{gauss}) - H(\mathbf{X}) \tag{1.56}$$

donde $\mathbf{X}_{gauss}$ es un vector aleatorio gaussiano con la misma matriz de covarianza $\underline{\mathbf{C}}_{\mathbf{X}}$ que $\mathbf{X}$. La entropía de dicho vector gaussiano se determina por

$$H(\mathbf{X}_{gauss}) = \frac{1}{2} \log |\det \underline{\mathbf{C}}_{\mathbf{X}}| + \frac{n}{2}(1 + \log 2\pi). \tag{1.57}$$

La negentropía posee una serie de propiedades que la hacen preferible a

la entropía, principalmente es invariante frente a transformaciones lineales:

$$J(\underline{\mathbf{M}}\mathbf{X}) = J(\mathbf{X}) \tag{1.58}$$

donde $\underline{\mathbf{M}}$ es una matriz de coeficientes reales.

1.2.14.5 Información Mutua

La información mutua se emplea como una medida de la cantidad de información que las variables de un vector aleatorio tienen sobre el resto de las variables del conjunto. Se puede expresar a partir del concepto de entropía como:

$$I(\mathbf{X} = [X_1, \ldots, X_n]) = \sum_{i=1}^{n} H(X_i) - H(\mathbf{X}). \tag{1.59}$$

A partir del corolario de la expresión (1.55) resulta obvio que la información mutua es siempre mayor o igual que cero, siendo cero si y sólo si las variables son independientes.

Respecto a su aplicación en el análisis de componentes independientes, la información mutua mide la dependencia estadística de varias variables aleatorias y da una idea de la información que comparten entre ellas. De tal modo, habría que minimizar la información mutua de las salidas para conseguir las componentes independientes.

1.3 Modelos de Mezcla Lineales

Podemos asumir que los datos de los que disponemos son un conjunto de variables que recibimos conjuntamente. Sea n el número de variables y T el

número de observaciones. Denotamos los datos disponibles como $x_i(t)$ donde los índices toman los valores $i = 1, \ldots, n$ y $t = 1, \ldots, T$. Suponemos, por simplicidad, que el número de observaciones, $x_i(t)$, es igual al número de fuentes, $s_i(t)$.

1.3.1 Modelo de mezcla lineal instantáneo

El modelo de mezcla instantánea y lineal (o llamado también lineal sin memoria) es el más sencillo de los propuestos, pero también y de momento, ha sido el más tratado por los autores. Este modelo supone que el valor de las observaciones ($x_i(t)$) en un instante dado es una función lineal de los valores de las fuentes originales ($s_i(t)$) en ese mismo instante de tiempo y no influyen en la mezcla en ningún caso los valores de las fuentes en instantes de tiempo anteriores. Una función lineal sobre un conjunto de variables se reduce a un conjunto de coeficientes reales que multiplican a los datos originales. De esta forma, se pude describir este modelo matemáticamente mediante la siguiente expresión:

$$x_i(t) = \sum_{j=1}^{n} a_{ij} s_j(t), \qquad i = 1, \ldots, n \tag{1.60}$$

o de forma resumida, utilizando vectores de variables:

$$\mathbf{x}(t) = \underline{\mathbf{A}}\mathbf{s}(t), \qquad \text{donde} \qquad \underline{\mathbf{A}} = \begin{bmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{1n} & \ldots & a_{nn} \end{bmatrix}. \tag{1.61}$$

La matriz $\underline{\mathbf{A}}$ se denomina matriz de mezcla, y es claramente la encargada

de realizar la combinación lineal de las fuentes $\mathbf{s}$. Al asumir que el número de observaciones o mezclas es igual al de fuentes, la matriz de mezcla $\underline{\mathbf{A}}$ es cuadrada, tal y como se muestra en la expresión (1.61).

Por tanto, el problema de la separación ciega de señales lineal e instantáneamente modelado, se podría resolver si se obtuviera una matriz $\underline{\mathbf{W}}$ llamada matriz de separación, cuya inversa es similar a la matriz de mezcla $\underline{\mathbf{A}}$ y de sus mismas dimensiones, con lo que, multiplicándola por las señales mezcladas se podría obtener una reconstrucción o estimación ($\mathbf{y}(t)$) de las fuentes originales. Naturalmente, es necesario que la matriz de separación $\underline{\mathbf{W}}$ sea no singular. De esta forma, la ecuación del modelo para la separación es:

$$\mathbf{y}(t) = \underline{\mathbf{W}}\mathbf{x}(t) \tag{1.62}$$

Reduciendo el problema de la separación ciega de señales lineal e instantánea mediante la aplicación del análisis de componentes independientes, recordemos que la solución será válida si las estimaciones $\mathbf{y}(t)$ son equivalentes a las fuentes $\mathbf{s}(t)$ salvo permutaciones y escalados invertibles:

La separación ciega de señales en medios lineales e instantáneos se reduce según la ecuación (1.61) a la obtención de los coeficientes de la matriz de separación $\underline{\mathbf{W}}$. Las dos indeterminaciones intrínsecas en el análisis de componentes independientes implican que $\underline{\mathbf{W}}$ es válida para recuperar las fuentes, aunque sus filas estén en distinto orden y multiplicadas por constantes arbitrarias, respecto a las filas de la matriz original de mezcla $\underline{\mathbf{A}}$.

Formalmente, diremos que la matriz de separación $\underline{\mathbf{W}}$ proporciona una

solución al problema de la separación ciega de señales que preserva la forma de las señales originales (aunque probablemente no el orden, ni sus magnitudes) si:

$$\underline{\mathbf{W}}\ \underline{\mathbf{A}} = \underline{\mathbf{\Lambda}}\ \underline{\mathbf{P}} \tag{1.63}$$

donde Λ es una matriz diagonal no singular y $\underline{\mathbf{P}}$ es una matriz de permutación (matriz con un único 1 por fila y columna).

Podemos definir una relación "$\sim$" denominada "relación de preservación de la forma de onda" entre dos matrices para el caso en que se cumple la expresión (1.63), es decir $\underline{\mathbf{W}}\ \underline{\mathbf{A}}$. Se puede demostrar que dicha relación es una relación de equivalencia [12]. En estos términos, el objetivo de la separación ciega de señales en el caso de mezclas lineales instantáneas es el de encontrar la clase de equivalencia de la cual $(\underline{\mathbf{A}}, \mathbf{s})$ forma parte. El modelo de mezcla y separación lineal e instantáneo se muestra gráficamente en la Figura 1.3.

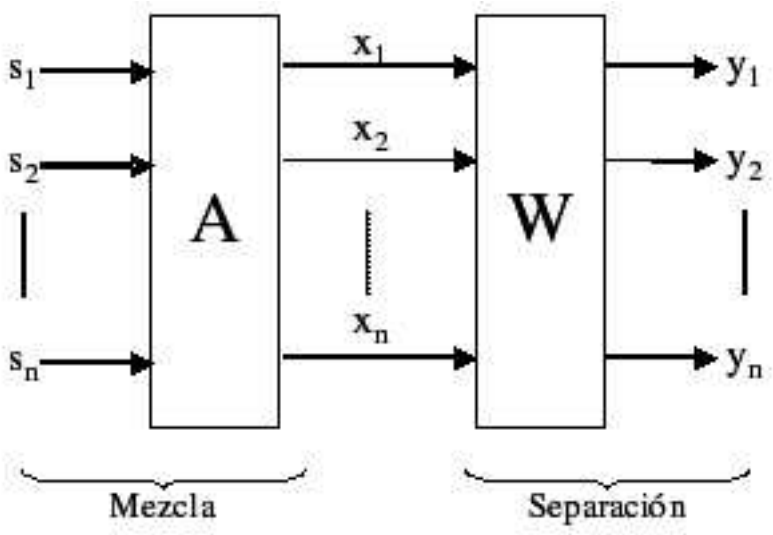

Figure 1.3: Modelo de mezcla de separación en el caso lineal e instantaeo.

1.3.2 Modelo de Mezcla Lineal Convolutivo

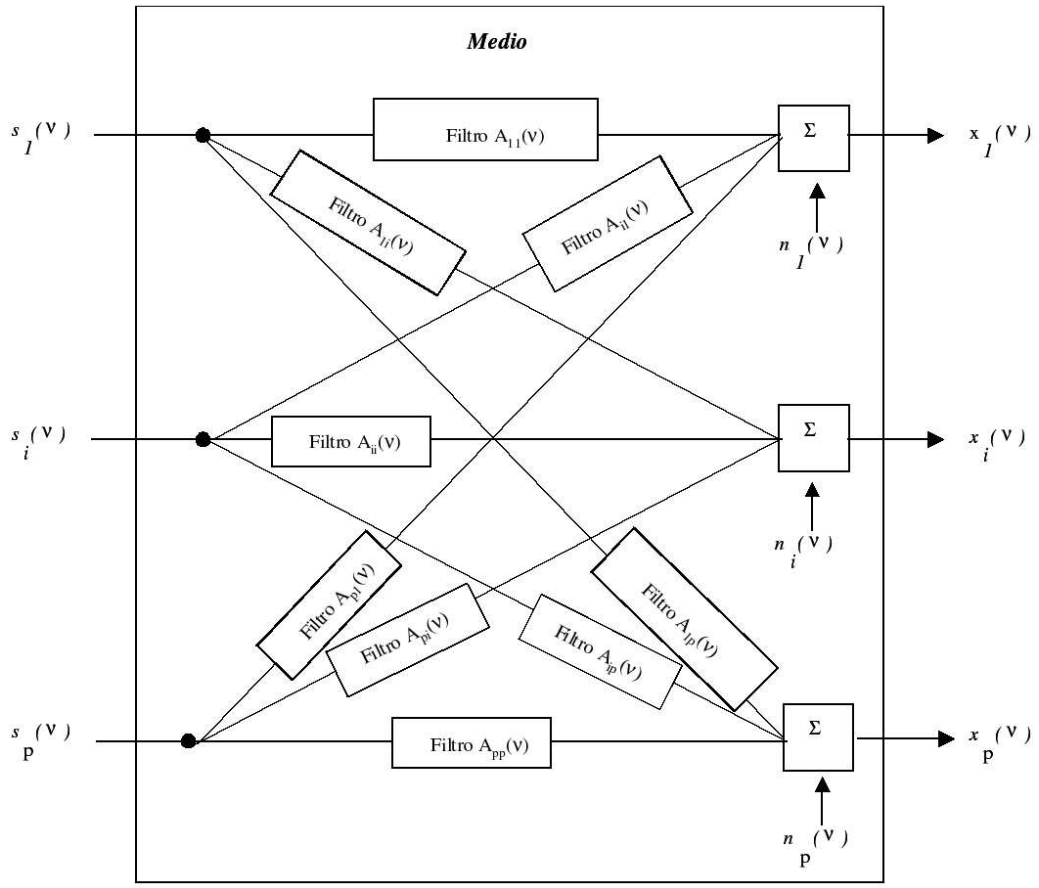

Figure 1.4: Modelo de mezcla convolutiva en el dominio de la frequencia para el caso en que el número de entradas y salidas coinciden.

En el modelo de mezcla convolutiva, llamada también mezcla con memoria o con retardos, se supone que el valor de las observaciones ($vecx_i(t)$) en un instante dado es una función de los valores de las fuentes originales ($\mathbf{s}_i(t)$) en ese mismo instante de tiempo y de valores de las fuentes en instantes de tiempo anteriores [67]. Matemáticamente, se podría representar este modelo en el caso general (lineal y no lineal) mediante una serie de funciones como

sigue:

$$\mathbf{x}(t) = F\left(\mathbf{s}(t), \mathbf{s}(t-1), \mathbf{s}(t-2), \ldots, \mathbf{s}(t-n)\right) \tag{1.64}$$

Esta ecuación trata de modelar fenómenos reales como ecos en comunicaciones y reverberaciones de las señales. Hay modelos de mezcla convolutiva [67] que consideran que el medio de propagación actúa como un filtro con p entradas (las fuentes) y q salidas (las señales detectadas por los sensores). Usualmente se admite que este filtrado, introducido por el medio y los sensores, es lineal y estacionario. De esta manera las señales captadas pueden expresarse de la forma:

$$x_i(t) = \sum_{j=1}^{p} \int a_{ij}(t-\tau) s_j(\tau) d\tau + n_i(t) \quad \forall\ i = 1, \ldots, q \tag{1.65}$$

donde $a_{ij}(t)$ representa la respuesta a un impulso de un filtro que modela la propagación entre la fuente i y el sensor j. El término $n_i(t)$ corresponde a un posible ruido aditivo, que no siempre tiene por qué estar presente.

En el dominio de la frecuencia, ν, la ecuación (1.64) puede escribirse como:

$$x_i(\nu) = \sum_{j=1}^{p} A_{ij} s_j(\nu) + n_i(\nu) \quad \forall\ i = 1, \ldots, q \tag{1.66}$$

donde la matriz $\underline{\mathbf{A}}(\nu)$, de dimensión $p \times q$, contiene todas las funciones de transferencia entre las p fuentes y los q sensores. De esta manera, el problema pasaría a ser lineal, pero en el dominio de la frequencia.

Los algoritmos de separación ciega de fuentes que se basan en el modelo de mezcla convolutiva tratan de explotar principalmente la llamada diversidad espacial que es el fenómeno que se produce por el hecho de la separación

física de los distintos sensores, lo que hace que cada uno de ellos capte, en cada instante de tiempo, distintas mezclas de las fuentes. Por otra parte, los modelos convolutivos se proponen con objeto de aprovechar la diversidad espectral, aunque el enfoque fundamental de la separación de fuentes es esencialmente espacial y busca la estructura a través de los sensores, no a lo largo del tiempo.

1.3.3 Modelos con Distinto Número de Mezclas y Fuentes

En aplicaciones reales, el número de fuentes puede ser desconocido e incluso variable en el tiempo. No obstante, la mayoría de los algoritmos de separación de señales son particularmente eficientes si el número de sensores es igual al número de fuentes, y parten, de hecho, de esta hipótesis. Si denominamos como p al número de fuentes de señal y por q al número de observaciones o de sensores, en caso de que no se verifique la igualdad $p = q$ hay dos posibilidades:

1. Más observaciones que fuentes ($q > p$):

   En este caso puede usarse una etapa previa de llamada de blanqueo 1.2.11 que transformará las q observaciones en las p señales incorreladas con mejor relación señal/ruido. Esta reducción dimensional se puede realizar con un Análisis de Componentes Principales (PCA) Primero se estima la matriz de covarianza de las observaciones y después se calculan sus valores propios. Las p señales proyectadas sobre los vectores propios correspondientes a los p valores propios mayores resultan ser las más adecuadas (las $q - p$ señales resultantes pueden considerarse

ruido).

2. Menos observaciones que fuentes ($q < p$):

   Esta situación se suele denominar separación ciega de señales sobrecompleta (overcomplete blind source separation). El problema resulta, en principio, irresoluble ya que las q observaciones no contienen información suficiente para separar exactamente las p fuentes. Se podría hacer un símil matemático de intentar resolver un sistema de ecuaciones con más incógnitas que ecuaciones. Se pueden estimar las p señales más potentes, pero con una distorsión provocada por las otras $p-q$ señales. Se ha estudiado teóricamente este problema consiguiendo separar las p fuentes en q grupos distintos y se obtuvieron las fuentes casi puras o combinaciones lineales de ellas . Se pueden encontrar algunos resultados experimentales con el algoritmo PCA-no lineal. Asimismo, según recientes investigaciones con métodos geométricos pueden detectarse las p componentes independientes [32], [89].

1.3.4 Mezclas no Estacionarias

Usualmente hemos supuesto que los elementos a_{ij} de la matriz de mezcla son constantes en el tiempo, lo que da lugar a las denominadas mezclas estacionarias. No obstante, puede ocurrir que las características del medio material donde se produce la mezcla varíen con el tiempo, o bien que la distribución de los sensores donde se captan las observaciones también varíe. Para modelar uno u otro fenómeno, los elementos de la matriz de mezcla A, han de cambiar en el tiempo ($\underline{\mathbf{A}} = \underline{\mathbf{A}}(t)$). También puede ocurrir que

las fuentes cambien, por ejemplo su densidad de probabilidad. Todos estos factores complican el problema de la separación ciega de señales. Por este motivo, es importante en la práctica que los algoritmos de separación puedan adaptarse a cambios de los coeficientes de la mezcla, e incluso de la distribución estadística de las fuentes.

Dadas las peculiaridades de las redes neuronales artificiales [73], [38], la utilización de éstas para implementar sistemas no estacionarios de separación de fuentes puede resultar muy eficiente. En efecto, es posible conseguir que los elementos de matriz estimada se auto-adapten en el tiempo a los cambios de la matriz de mezcla, aplicando constantemente la regla de aprendizaje y seleccionando adecuadamente el parámetro de ganancia de aprendizaje en el tiempo. Algunos autores [60], [67] proponen una red neuronal que considera la no estacionalidad de las fuentes. Cichocki et al. [22], [22] proponen un algoritmo auto-adaptativo para entornos no estacionarios, basado en la adecuada modificación de la ganancia de aprendizaje a lo largo del tiempo.

1.3.5 Otros Modelos de Mezcla Lineales

1.3.5.1 Presencia de Ruido

El ruido suele considerarse, dentro del problema de separación de fuentes, bajo tres perspectivas distintas:

1. El ruido incide en el medio como una señal más. En este caso los algoritmos que se proponen consideran al ruido como una de las p señales y lo representan correctamente como si se tratara de una fuente cualquiera, realizando la separación, si es posible.

2. Las señales originales están afectadas por ruido. En esta situación los algoritmos de separación actúan transparentemente con respecto al ruido, sin incrementarlo ni reducirlo, recuperándose las señales con la misma relación señal-ruido original.

3. El ruido corrompe las señales ya mezcladas, en el interior del medio o de los sensores. Este es un caso en el que el ruido debe ser tenido en cuenta por las técnicas de separación para reducirlo y, si es posible, eliminarlo. Para modelar este caso, se suele considerar que el ruido es aditivo y acotado [25],[47], [49].

La mayoría de los autores consideran el ruido como una fuente desconocida más, denotándolo a veces por $\mathbf{n}(t)$. En cualquier caso se observa que la presencia de ruido empeora notablemente las prestaciones de los algoritmos de separación, complicándolos a su vez. Lo ideal sería disponer se un modelo de separación ciega de fuentes que permitiese eliminar totalmente el ruido en las observaciones. J.F. Cardoso [17] afirma que es una cuestión abierta determinar los dominios de aplicación donde realmente es de interés considerar modelos de ruido. Es usual modelar el ruido, $\mathbf{n}(t)$, mediante [64]:

1. una señal aleatoria con densidad de probabilidad uniforme. Este ruido se denomina ruido aleatorio uniforme o ruido subgaussiano.

2. una señal con densidad de probabilidad normal o gaussiana. Este ruido se denomina ruido aleatorio gaussiano.

3. una señal con densidad de probabilidad laplaciana. Este ruido se denomina ruido aleatorio supergaussiano.

1.3.6 Conocimiento de las Fuentes

Cualquier información sobre las fuentes puede dar lugar a simplificaciones en el algoritmo de separación o a mejorar notablemente los resultados. En este caso, ya no se podría hablar de separación ciega de fuentes de manera estricta, puesto que disponemos de un conocimiento concreto sobre algún aspecto de las fuentes. En cuanto a la información que se puede tener sobre las fuentes esta puede ser:

1. Conocimiento sobre las distribuciones de probabilidad de las fuentes [9], [10], [74]

2. Conocimiento sobre la naturaleza discreta de los datos (binarios o multivaluados), en cuyo caso se pueden utilizar algoritmos muy eficientes[75], [77], [76].

3. En general saber también que las fuentes estén acotadas (hipótesis comúnmente aplicada, ya que responde a la realidad) y conocer los rangos máximo y mínimo entre los cuales pueden estar limitadas en amplitud [80],[79].

1.4 Modelos de Mezcla no Lineales

Existen muchas situaciones en que el modelo de mezcla lineal no es el apropiado para modelar fenómenos no lineales y resulta ser una aproximación demasiado alejada del modelo real. Como ejemplos prácticos de este tipo de situaciones, encontramos los ya citados canales que incorporan reverberaciones o ecos 1.3.2, canales de satélite digital y microondas compuestos de

un filtro lineal seguido de un amplificador no lineal instantáneo [88], canales de grabación magnética, etc.

Si generalizamos el modelo de mezcla lineal instantánea, suponiendo que el valor de las observaciones en un instante dado es una función de cualquier tipo que depende única y exclusivamente de los valores de las fuentes originales en ese mismo instante de tiempo, el modelo de mezcla será no lineal e instantáneo. Podemos generalizar la expresión (1.61) de forma que el medio de mezcla no sea lineal:

$$\mathbf{x}(t) = \mathbf{F}(\mathbf{s}(t)) \tag{1.67}$$

donde $\mathbf{F} = [f_1, f_2, \dots, f_n]^T$ es una familia de funciones no lineales de mezcla, que transforma las señales originales desconocidas en las mezclas observadas.

Si bien los modelos de mezcla lineal instantáneo, así como el convolutivo, han sido analizados y estudiados intensivamente, la generalización a modelos no lineales tan sólo ha sido estudiada en situaciones muy concretas, probablemente debido a los problemas de complejidad y multidimensionalidad inherentes al modelo.

El problema de la separación ciega de señales no lineal, consiste en encontrar una familia de funciones $\mathbf{G} = [g_1, g_2, \dots, g_n]^T$ tales que, aplicadas sobre las mezclas observadas, nos devuelvan un conjunto de señales $\mathbf{y}$ que sean equivalentes en su forma de onda a las señales originales $\mathbf{s}$. Matemáticamente el modelo de separación en el caso no lineal corresponde a obtener:

$$\mathbf{y}(t) = \mathbf{G}(\mathbf{x}(t)) \tag{1.68}$$

Como es lógico, en el mejor de los casos debemos encontrar un conjunto de

funciones $\mathbf{G}$ que sean las funciones inversas de $\mathbf{F}$, de forma que $\mathbf{G}(\mathbf{x}(t)) = \mathbf{s}(t)$ y podemos recuperar las fuentes originales:

$$\mathbf{G}(\mathbf{x}) = \mathbf{F}^{-1} \qquad \rightarrow \qquad g_i(x) = f_i^{-1}(x) \ \forall \ i = 1, \ldots, n. \tag{1.69}$$

1.4.1 El análisis de Componentes Independientes y la Separación Ciega de Señales en Mezclas no Lineales

El objetivo del análisis de componentes independientes es encontrar un transformación de un conjunto de variables independientes, de forma que las variables transformadas sean independientes. Se ha mostrado que el problema de la separación ciega de señales en el caso de mezclas lineales, tanto instantáneas como algunas de sus variantes, se reduce a encontrar las componentes independientes, que serán equivalentes a las fuentes salvo en posibles permutaciones y escalados invertibles (Sección 1.1.2).

Si asumimos un modelo de mezcla no lineal, encontrar n componentes independientes no necesariamente proporciona un conjunto de señales equivalentes a las fuentes desconocidas. En este caso, el objetivo del análisis de componentes independientes es encontrar una aplicación $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ tal que $\mathbf{y} = \mathbf{g}(vecx)$ sea independiente.

Como veremos en el siguiente apartado, la característica fundamental del análisis de componentes independientes no lineal es que, en el caso general (1.67), siempre existen soluciones y que éstas son altamente no únicas

[47],[48]. Es decir, existen multitud de soluciones diferentes que, a partir de un conjunto de variables dependientes, nos proporcionan un conjunto de variables independientes. El problema reside en que dichas soluciones pueden no tener ninguna relación con las señales originales que estamos buscando. Por esta razón, decimos que en el caso general de mezclas no lineales, los problemas de ICA y separación ciega de señales no coinciden (tal como sucedía para mezclas lineales).

1.4.2 Resultados de Existencia y Unicidad del Análisis de Componentes Independientes no Lineal

Varios autores [47],[35],[91] han obtenido recientemente resultados sobre la existencia y unicidad de soluciones para ICA y separación ciega de señales no lineales; llegando a la conclusión de que el problema del análisis de componentes independientes en el caso de mezclas no lineales se caracteriza por:

1. Existencia: es posible construir una función $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^n$ tal que $\mathbf{y} = \mathbf{g}(\mathbf{x})$ sea independiente.

2. No unicidad: es posible construir un número infinito de funciones $\mathbf{g} : \mathbb{R}^n \to \mathbb{R}^n$ tal que $\mathbf{y} = \mathbf{g}(\mathbf{x})$ sea independiente, aparte de funciones de permutación y escalados invertibles.

Si bien Hyvärinen y Pajunen [47] demuestran formalmente ambas características a partir del análisis de factores, queda claro que si X e Y son dos variables aleatorias independientes, cualquier función de éstas variables $f_X(x)$ y $g_Y(y)$ también lo será.

En cualquier caso, parece obvio que en el caso de mezclas no lineales necesitaremos información adicional acerca de las señales originales o del medio de mezcla para poder restringir el conjunto infinito de soluciones que nos proporciona la técnica de ICA-no lineal a un conjunto reducido de soluciones equivalentes a las señales originales. En caso de no disponer de información adicional, la separación de señales mediante el análisis de componentes independientes es, simplemente, imposible.

1.4.3 El Modelo de Mezcla Post-no-lineal

El modelo post-no-lineal (PNL) corresponde a un caso concreto de la mezcla no lineal expresada en la ecuación (1.67) en el que se supone que cada una de las fuentes desconocidas ha sido mezclada en un proceso de dos fases:

1. Mezcla lineal: las fuentes $\mathbf{s}$ se mezclan en primer lugar de acuerdo al modelo de ICA lineal básico 1.61, obteniendo un conjunto de mezclas también desconocidas *vecz*:

$$\mathbf{z}(t) = \underline{\mathbf{A}}\mathbf{s}(t). \tag{1.70}$$

2. Mezcla no lineal: se aplica una función no lineal f_i en cada uno de los canales z_i, obteniendo las observaciones o mezclas finalmente disponibles:

$$x_i(t) = f_i z_i(t) \tag{1.71}$$

En la Figura 1.5 se puede observar en la parte izquierda el modelo de mezcla en dos fases. En resumen, la ecuación del modelo de mezcla en el

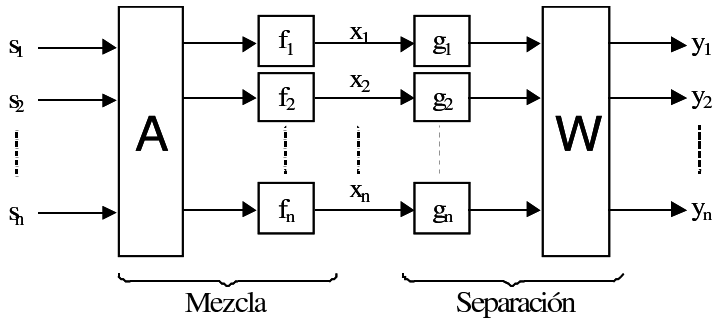

Figure 1.5: Modelo de mezcla y separación post-no-lineal.

caso post-no-lineal para cada una de las componentes es:

$$x_i = f_i\left(\sum_{j=1}^{n} a_{ij}s_j\right) \tag{1.72}$$

Que podemos rescribir para vectores de variables como:

$$\mathbf{x} = \mathbf{F}(\underline{\mathbf{A}}\mathbf{s}) \tag{1.73}$$

donde $\underline{\mathbf{A}}$ es la matriz de mezcla de coeficientes lineales y $\mathbf{F} = (f_1, f_2, \ldots, f_n)$ la familia de distorsiones no lineales.

Aparte del interés teórico que pueda tener este modelo, se adapta perfectamente a multitud de problemas reales en los que, si bien la interacción del medio con las fuentes se puede modelar mediante una mezcla lineal (primera fase), son los propios sensores o receptores los que incorporan el carácter no lineal de la mezcla (segunda fase), en forma de distorsiones o saturaciones de la señal. Este tipo de casos se suele producir en aplicaciones de grabaciones magnéticas, comunicaciones vía satélite, etc.

Se puede demostrar [93], que para este modelo de mezcla post-no-lineal las indeterminaciones encontradas aplicando la técnica de análisis de componentes independientes (ICA) son las mismas que en el caso de separación ciega de señales en mezclas lineales e instantáneas (Apartado 1.3.1). De esta forma, ICA y el problema de la separación ciega de señales vuelven a coincidir, siendo la solución obtenida única y equivalente a las fuentes originales salvo las mencionadas permutaciones y cambios de escala. Consecuentemente, la ecuación de separación para este modelo es la siguiente:

$$y_i = \sum_{j=1}^{n} b_{ij} g_j(x_j) \tag{1.74}$$

O en modo vectorial:

$$\mathbf{y} = \underline{\mathbf{W}}\ \underline{\mathbf{G}}(\mathbf{x}) \tag{1.75}$$

donde $\underline{\mathbf{W}}$ es la matriz de separación de coeficientes lineales y $\underline{\mathbf{G}} = (g_1, g_2, \ldots, g_n)$ la familia de funciones inversas para deshacer las no-linealidades. En la Figura 1.5 se puede observar en la parte derecha el proceso de estimación o separación de las fuentes originales. No obstante, el problema de encontrar las componentes independientes es computacionalmente mucho más complejo que en el caso lineal, debido a que el espacio de búsqueda es mucho mayor, por lo que es muy complicado encontrar soluciones satisfactorias conforme crece la dimensionalidad del problema.

1.5 Aplicaciones de la Separación ciega de señales

Al presentarse una transmisión de información desde un emisor a un receptor a través de un canal de comunicación se puede producir una alteración de la información transmitida, lo cual puede conducir a consecuencias no deseadas. Entre este amplio margen de aplicaciones se pueden encontrar diferentes campos de la ciencia y la técnica, como los que seguidamente describimos.

1.5.1 Supresión de Ruido e Interferencias

La separación de señales permite la extracción de fuentes de ruido para disponer de señales útiles en diversas aplicaciones. Desde hace tiempo, se han estudiado y detectado numerosos tipos de ruido en diversos dispositivos [36]; por ejemplo, el producido por efecto Shot (transporte), el generado por efecto Flicker (modulación), el ruido térmico (fluctuaciones), el ruido Johnson, el ruido impulsivo (atmosférico y solar) y el ruido cuántico (optoelectrónica, superconductores, dispositivos de electrones calientes). Por otra parte, las fuentes de ruido pueden ser internas, presentes en los conductores, semiconductores, radiorreceptores, diodos, amplificadores, osciladores, transistores, circuitos, TV y generadores de pulsos, entre otros, y externas, afectando a antenas, comunicaciones, receptores de UHF, scattering, sistemas no lineales, comunicaciones de pulsos, sonar, transmisión de datos binarios, modulación PCM, radar y radionavegación, entre otros.

Los sistemas con conjuntos de micrófonos están siendo utilizados en cap-

tura de la voz, localización de fuentes acústicas y en aplicaciones tales como teléfonos móviles, video-conferencias e interfases de audio con sistemas basados en Computadores Personales (PCs). Los inconvenientes de estos sistemas son la dependencia de la geometría del conjunto de micrófonos, las características acústicas del medio y la degradación de las señales debida a la reverberación, así como las prestaciones no lineales de los sensores acústicos. En este ámbito, otras alternativas como el filtrado adaptativo y la sustracción espectral están siendo muy eficientes [69].

1.5.2 Aplicaciones Biomédicas

En relación con las señales biomédicas, la separación de fuentes es una técnica prometedora en este campo, dado que los métodos tradicionales de identificación y medida de respuestas se basaban en las amplitudes y latencias de los picos de la forma de onda y en modelos de dipolos múltiples que proporcionaban resultados ambiguos cuando la geometría de las fuentes es desconocida o compleja [57]. De hecho, existen en la actualidad líneas de investigación para aplicaciones de tratamiento de señal en biomedicina relacionadas con el registro de señales y con la captación de varias señales con multisensores. Además, en la captación de señales biomédicas se usan varios sensores situados próximos unos de otros, con el inconveniente de que no sólo se graban las señales deseadas sino también otras procedentes de otros fenómenos biológicos, lo que proporciona un ambiente idóneo para aplicar la separación de fuentes. Es interesante, por ejemplo utilizar métodos de Separación Ciega de Fuentes para separar, a partir de Electroencefalogramas

(EEG), señales de distinta naturaleza (llamadas artefactos) como pueden ser el movimiento de los ojos, ruido producido por los músculos, o el latido del corazón de las señales propiamente encefalográficas [8].

El análisis de imágenes obtenidas por Resonancia Magnética Nuclear (MNR) permite controlar a una persona mientras realiza tareas psicomotrices, sin embargo estas lecturas suelen ir contaminadas por otras señales lo que impide un buen seguimiento de las que interesan. En este campo también se ha aplicado el análisis de componentes independientes [54] para buscar una solución. Dentro del mismo campo se encuentra la reconstrucción ciega de flujo autónomo cardiaco en el rango de bajas frecuencias (LF) usando solo el ritmo cardiaco y la presión arterial de la sangre [94]. Otras aplicaciones de interés son la descomposición de potenciales evocados obtenidos con un magnetoencefalógrafo (MEG) producidos después de una estimulación sensorial [95], la separación de patrones de señales espaciales a partir de grabaciones de imágenes ópticas [83], separación de señales procedentes de un Electrocardiograma (ECG) del corazón de una mujer embarazada y su feto [50].

1.5.3 Aplicaciones en Audio

Uno de los orígenes de la separación de fuentes se basó en encontrar una explicación al efecto "cocktail party" [37]. Este efecto consiste en la habilidad del ser humano de poder centrar su atención y escuchar a una sola persona, aún habiendo más personas hablando a la vez o existiendo otras fuentes sonoras. Dicha especial capacidad de escucha selectiva se explica debido al procesamiento de alto nivel del sonido y el lenguaje que tiene lugar

en el sistema nervioso central, más que a las características propias del sistema auditivo humano [13]. Por tanto, una de las primeras aplicaciones que surgieron fue la de separar señales de audio, voz por ejemplo, en distintos entornos.

1.5.4 Aplicaciones en telecomunicaciones

Otro campo interesante de aplicación de la Separación Ciega de Fuentes es el de las telecomunicaciones, donde también tiene cabida la aplicación de la eliminación de ruido, existiendo una demanda creciente de aplicaciones controladas por la voz que exhiban prestaciones robustas e independientes del medio. Dentro de este campo se podrían enumerar algunas aplicaciones como las siguientes:

- Tratamiento de antenas. Se parte de las señales captadas por una serie de antenas producida al incidir varias señales de banda estrecha originadas por fuentes situadas en lugares desconocidos. De nuevo lo que interesa es obtener aisladas dichas fuentes [19].

- Radar. Donde la eliminación de ruido es primordial [50].

- Comunicaciones móviles. Igual que en la anterior, la eliminación de ruido juega un gran papel [33].

- Separación de mensajes en sistemas de comunicación de múltiple acceso con división de código (CDMA) [78].

- Enlaces de alta frecuencia (HF) [20].

- Señales de identificación por radiofrecuencia de blancos múltiples utilizando separación de señales con redes neuronales artificiales [29] [30].
- Separación de señales captadas/generadas por satélites de telecomunicaciones [12].

1.5.5 Aplicaciones en Imagen

- Eliminación del ruido en imágenes y procesamiento, en general, de imágenes [43] [23] [82].[68]
- Identificación de los componentes de una mezcla de sustancias químicas a partir de una espectroscopia de Resonancia Magnética Nuclear (MNR).
- Clasificación de imágenes en medicina [8].

1.5.6 Otras Aplicaciones

Existen otros muchos campos de aplicación, entre los cuales a continuación se enumeran algunos ejemplos:

- Separación de dos fuentes ópticas infrarrojas [48].
- vigilancia de instalaciones [**?**].
- Control de máquinas eléctricas rotativas. En una sala con gran cantidad de máquinas eléctricas se puede detectar el sonido individual producido por cada una de ellas, y localizar si se produce alguna avería en alguna máquina [35] [98].
- Análisis de datos sísmicos [90].

- clasificación no supervisada [54].

- Sensores inteligentes. En sensores eléctricos y magnéticos la señal a detectar suele estar enmascarada con señales de otro origen, usualmente térmico. Se trata de separar automáticamente la influencia de las distintas fuentes [70].

Chapter 2

ICA algorithms

In this chapter, some of the most important ICA algorithm will be presented.

2.1 fastICA

A very popular ICA algorithm, which has already been applied successfully to many real world problems is fastICA[44]. For this algorithm the observations have to have zero mean as well as a diagonal covariance matrix. The latter condition can be fulfilled if the observations are whitened. For this purpose, the eigenvalue decomposition of covariance matrix of the observations $\underline{\mathbf{C}}$ has to be computed:

$$\underline{\mathbf{C}} = \underline{\mathbf{V}}\ \underline{\mathbf{D}}\mathbf{V}^T \tag{2.1}$$

where $\underline{\mathbf{V}}$ is the eigenvector matrix and $\underline{\mathbf{D}}$ is a diagonal matrix containing the eigenvalues of $\underline{\mathbf{X}}$ on its diagonal. Based on the matrices $\underline{\mathbf{D}}$ and $\underline{\mathbf{V}}$, the

whitening matrix $\underline{\mathbf{Q}}$ is determined as follows:

$$\underline{\mathbf{Q}} := \underline{\mathbf{D}}^{-1/2}\underline{\mathbf{V}}^T \tag{2.2}$$

Multiplying this matrix with the observation matrix $\underline{\mathbf{X}}$ leads to the desired uncorrelatedness of the observations.

In the sequel, it will be assumed that the observations are zero mean and white.

The goal of the fastICA algorithm is to find a matrix $\underline{\mathbf{W}}$ such that the rows of $\underline{\mathbf{Y}} = \underline{\mathbf{W}}\,\underline{\mathbf{X}}$ are statistically independent. In fastICA the independence of the components of a random vector is measured by means of the mutual information I. This independence measure can be expressed in terms of the negentropy J as follows:

$$I(y_1, y_1, \ldots, y_N) = J(\mathbf{y}) - \sum_i J(y_i). \tag{2.3}$$

I is a nonnegtive function which vanishes if the elements of $\mathbf{y}$ are statistically independent. As the negentropy $J(\mathbf{y})$ is independent under linear transformations (i.e. $J(\underline{\mathbf{W}}\mathbf{x}) = J(\mathbf{x})$ for some matrix $\underline{\mathbf{W}}$ and some vector $\mathbf{x}$) only the second term in the above equation needs to optimized in the context of ICA.

Hence, the fastICA problem can be stated as follows

$$\text{Maximize } \sum_{i=1}^{N} J(\mathbf{w}_i) \text{ w.r.t. } \mathbf{w}_i, i = 1, \ldots, n \tag{2.4}$$

$$\text{under the constraint } E\{(\mathbf{w}_k^T\mathbf{x})(\mathbf{w}_j^T\mathbf{x})\} = \delta_{jk}.$$

as $y_i = \mathbf{w}_i^T \underline{\mathbf{X}}$. The normalization $E\{(\mathbf{w}_k^T\mathbf{x})(\mathbf{w}_j^T\mathbf{x})\} = \delta_{jk}$ is necessary as the underlying sources are assumed to be statistically independent, an hence also uncorrelated, in ICA.

For the optimization task a good estimate of the negentropy of a random variable is needed. In [44] the following estimation formula is used:

$$J(y_i) = \frac{1}{\delta^2}(E\{G(y_i)\} - E\{G(\nu)\})^2. \tag{2.5}$$

Here, G is any nonlinear function, ν is a standardized gaussian variable and δ is a normalizing constant.

This expression for $J(y_i)$ can be optimized under the given constraints by means of the Kuhn-Tucker conditions and the Newton method. After some lenghty calculations (see [44] for a detailed derivation) this leads ot the following update rule for the individual rows $\mathbf{w}$ of $\underline{\mathbf{W}}$:

$$\mathbf{w}^+ = E\{\mathbf{x}G'(\mathbf{w}^T\mathbf{x})\} - E\{G''(\mathbf{w}^T\mathbf{x})\}\mathbf{w} \tag{2.6}$$

$$\mathbf{w}^* = \frac{\mathbf{w}^+}{||\mathbf{w}^+||} \tag{2.7}$$

This procedure is used iteratively until all rows of $\underline{\mathbf{W}}$ are determined. However, it must be avoided that the algorithm converges several times to the same maxima of the negentropy. This can be achieved, if after every iteration of the Newton method the newly determined row vector $\mathbf{w}_{q+1}$ is decorrelated with respect to the already determined rows $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_q$ of $\underline{\mathbf{W}}$.

Once the matrix $\underline{\mathbf{W}}$ has been determined, the sources can be obtained by

simply multiplication the inverse of $\underline{\mathbf{W}}$ with the observations.

Chapter 3

Nonnegative Matrix Factorization and Sparseness

So far only BSS algorithms have been discussed which are based on the assumption that the underlying sources are mutually independent. Even if this assumption is very attractive with regard to the uniqueness of the obtained results it is still hard to justify in the context of microarray data. Hence, in this section alternative matrix factorization approaches will be presented in which the assumptions made on $\underline{\mathbf{A}}$ and $\underline{\mathbf{S}}$ are closer related with the biological processes occuring in cells.

One way to incorporate such biological preknowledge is to require that the matrices $\underline{\mathbf{A}}$ and $\underline{\mathbf{S}}$ obtained by the matrix factorization approaches are nonnegative. As will be discussed later, this assumption is reasonable as the activity of a cellular process cannot be negative and as only nonnegative amounts of mRNA can be produced by cells. Furthermore, biological organisms always try to save energy by synthesizing as small amounts of mRNA

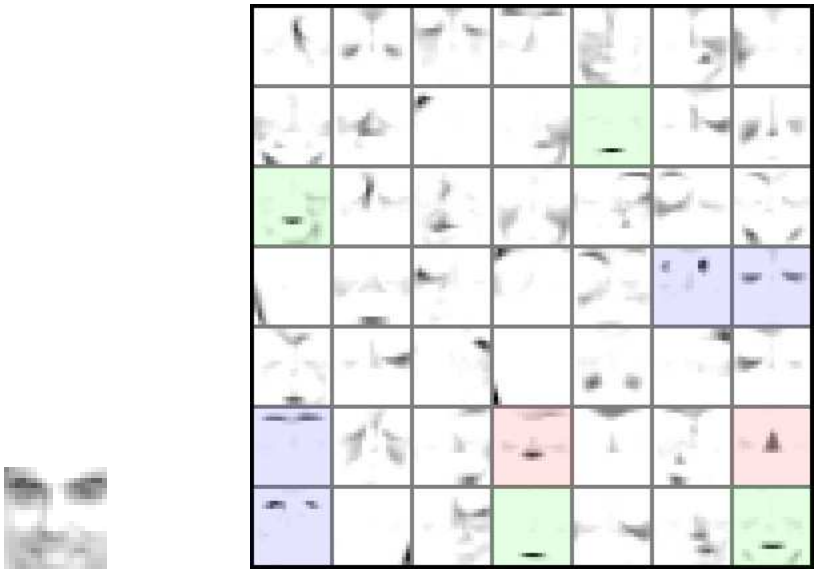

Figure 3.1: Face image decomposition by NMF. Left: example of an original image taken from the CBCL database. Right: basis images (rows of $\underline{\mathbf{S}}$) as obtained by NMF of the face image data from CBCL (figure reproduced from [41]). As can be seen the faces are decomposed into typical parts: nose (highlighted in pink), eyes (light blue) and mouth (turquoise).

as possible. Accordingly, the rows of the matrix $\underline{\mathbf{S}}$ will have many nil entries, a property which can also be exploited explicitly in matrix factorization approaches.

In the following, one of the most important non negative matrix factorization (NMF) algorithms will be reviewed. Furthermore, it will be shown how sparseness constraints can be incorporated into this algorithm.

3.1 Nonnegative Matrix Factorization

In nonnegative matrix factorization, a nonnegative observation matrix $\underline{\mathbf{X}}$ is decomposed into two nonnegative matrices $\underline{\mathbf{A}}$ and $\underline{\mathbf{S}}$

$$\underline{\mathbf{X}} = \underline{\mathbf{A}}\ \underline{\mathbf{S}},$$

whereas $\underline{\mathbf{X}}$ is an $M \times T$, $\underline{\mathbf{A}}$ is an $M \times N$ and $\underline{\mathbf{S}}$ is an $N \times T$ matrix.

In contrast to the independence assumption used in ICA, the nonnegativity constraints for the matrices $\underline{\mathbf{A}}$ and $\underline{\mathbf{S}}$ are insufficient to obtain unique results. This can easily be seen as for instance

$$\underline{\mathbf{X}} = \underline{\mathbf{I}}\ \underline{\mathbf{X}}, \tag{3.1}$$

where $\underline{\mathbf{I}}$ denotes the identity matrix, is always a valid nonnegative factorization of $\underline{\mathbf{X}}$. Still, NMF has lead to useful and meaningful results in many research areas like in image analysis, hyperspectral data processing, biological modeling and sparse coding. Thereby, the obtained features (the rows of the matrix $\underline{\mathbf{S}}$) are often sparsely represented, i.e. contain many nil entries. This sparsity often leads to easily interpratable results as was shown, for instance, in [52]. There, NMF was applied to face image data from the CBCL database [1]. As can be seen in Fig. 3.1, these faces could be decomposed into their major parts like, for instance, the nose, eyes or lips.

3.1.1 NMF algorithm

Especially in applications where the number of observations M is very large, N is often chosen to be smaller than M in order to obtain a compressed version of the original data matrix. Obviously, only an approximative factorization of $\underline{\mathbf{X}}$ can be obtained in such cases

$$\underline{\mathbf{X}} \approx \underline{\mathbf{A}}\ \underline{\mathbf{S}}. \tag{3.2}$$

such that the goal of NMF can be stated as finding the best approximate factorization of the matrix $\underline{\mathbf{X}}$. Accordingly, a cost function is needed to quantify the quality of the approximation. For this purpose, the square of the Euclidean distance between two matrices $\underline{\mathbf{C}}$ and $\underline{\mathbf{D}}$ is often used:

$$\|\underline{\mathbf{C}} - \underline{\mathbf{D}}\|^2 = \sum_{ij}(C_{ij} - D_{ij})^2. \tag{3.3}$$

Based on this quality measure the NMF optimization problem can be formulated at follows:

$$\text{Minimize } \|\underline{\mathbf{X}} - \underline{\mathbf{A}}\ \underline{\mathbf{S}}\|^2 \text{ w.r.t. } \underline{\mathbf{A}} \text{ and } \underline{\mathbf{S}}, \text{subject to } \underline{\mathbf{A}}, \underline{\mathbf{S}} \geq 0. \tag{3.4}$$

Note, that although the function $\|\underline{\mathbf{X}} - \underline{\mathbf{A}}\ \underline{\mathbf{S}}\|$ is convex in $\underline{\mathbf{A}}$ only or $\underline{\mathbf{S}}$ only, it is not convex in both variables together. Hence, any optimization methods based on gradient descent are likely to find only local minima of the target function in (3.4).

Currently, the algorithm [51] propsed by Lee and Seung is one of the most

widely used approaches to solve the NMF problem as stated in (3.4). This algorithm is based on the following theorem:

Theorem 1. *The Euclidean distance* $\|\underline{\mathbf{X}} - \underline{\mathbf{AS}}\|$ *is nonincreasing under the update rules*

$$\begin{aligned} A_{ij} &\leftarrow A_{ij} \frac{(\underline{\mathbf{X}}\ \underline{\mathbf{S}}^T)_{ij}}{(\underline{\mathbf{A}}\ \underline{\mathbf{S}}\underline{\mathbf{S}}^T)_{ij}} \\ S_{ij} &\leftarrow S_{ij} \frac{(\underline{\mathbf{A}}^T\ \underline{\mathbf{X}})_{ij}}{(\underline{\mathbf{A}}^T\ \underline{\mathbf{AS}})_{ij}}. \end{aligned} \tag{3.5}$$

The Euclidean distance is invariant under these updates if and only if $\underline{\mathbf{A}}$ *and* $\underline{\mathbf{S}}$ *are at a stationary point of the distance.*

In other words, the new values for the entries of $\underline{\mathbf{A}}$ and $\underline{\mathbf{S}}$ are obtained by multiplicating their predecessors by a factor. As can be seen directly in (3.5) this factor is unity if $\underline{\mathbf{X}}$ is perfectly factorized, i.e. if $\underline{\mathbf{X}} = \underline{\mathbf{A}}\ \underline{\mathbf{S}}$.

The multiplicative update rules in (3.5) can be derived from the additive update rules used in gradient descent if a suitable step size parameter is chosen. To illustrate this, consider the additive update rule for $\underline{\mathbf{S}}$ that reduces the squared distance in (3.4) for a given $\underline{\mathbf{A}}$:

$$S_{ij} \leftarrow S_{ij} + \eta_{ij} \left[(\underline{\mathbf{A}}^T\ \underline{\mathbf{X}})_{ij} - (\underline{\mathbf{A}}^T \underline{\mathbf{A}}\ \underline{\mathbf{S}})_{ij} \right]. \tag{3.6}$$

If all η_{ij} are set to the same small positive number τ this update rule is equivalent to that applied in gradient descent. This means, that as long as τ is chosen to be sufficiently small, this update will reduce $\|\underline{\mathbf{X}} - \underline{\mathbf{AS}}\|$.

In order to obtain the multiplicative update rule for $\underline{\mathbf{S}}$ as shown in (3.5)

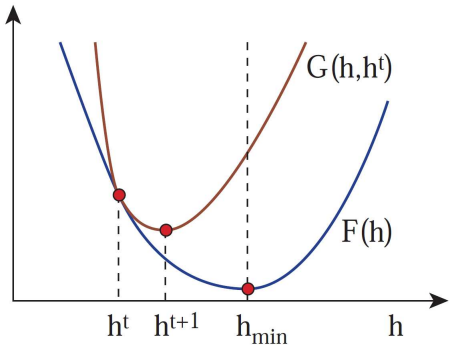

Figure 3.2: Minimization of $F(\mathbf{s})$: For $\mathbf{s}^{t+1} = \arg\min_{\mathbf{s}} G(\mathbf{s}, \mathbf{s}^t)$ minimizing the auxiliary function $G(\mathbf{s}, \mathbf{s}^t) \leq F(\mathbf{s})$ guarantees that $F(\mathbf{s}^{t+1}) \leq F(\mathbf{s}^t)$ (figure reproduced from [51]).

individual step sizes have to be computed for each element S_{ij} of $\underline{\mathbf{S}}$ as follows

$$\eta_{ij} = \frac{S_{ij}}{(\underline{\mathbf{A}}^T \, \underline{\mathbf{A}} \, \underline{\mathbf{S}})_{ij}}. \tag{3.7}$$

Even if these step sizes are not necessarily small it still can be shown rigorously that they always lead to a decreasing cost function. This will be proven in the next section.

3.1.2 Proof of convergence

To proof Theorem 1 of the last section so-called auxiliary functions $G(\mathbf{s}, \mathbf{s}')$ are needed. Such functions are defined as follows:

Definition 1. *$G(\mathbf{s}, \mathbf{s}')$ is an auxiliary function for $F(\mathbf{s})$ if $G(\mathbf{s}, \mathbf{s}')$ is continuous and if the conditions*

$$G(\mathbf{s}, \mathbf{s}') \geq F(\mathbf{s}), \qquad G(\mathbf{s}, \mathbf{s}) = F(\mathbf{s}) \tag{3.8}$$

are satisfied.

For these auxiliary functions the following lemma can be formulated which makes them an ideal tool to prove the convergence of the objective function F (see also Fig. 3.2):

Lemma 1. *If G is an auxiliary function, then F is nonincreasing under the update*

$$\mathbf{s}^{t+1} = \arg\min_{\mathbf{s}} G(\mathbf{s}, \mathbf{s}^t) \tag{3.9}$$

Proof. $F(\mathbf{s}^{t+1}) \leq G(\mathbf{s}^{t+1}, \mathbf{s}^t) \leq G(\mathbf{s}^t, \mathbf{s}^t) = F(\mathbf{s}^t)$ □

Obviously, if $\mathbf{s}^t$ is a local minimum of $G(\mathbf{s}, \mathbf{s}^t)$ then $F(\mathbf{s}^{t+1}) = F(\mathbf{s}^t)$. Furthermore, if $F(\mathbf{s})$ is differentiable and if its derivates are continuous in a small neighborhood of $\mathbf{s}^t$, also $\nabla F = 0$ must hold as otherwise $F(\mathbf{s}) > G(\mathbf{s}, \mathbf{s}^t)$ in the neighborhood of $\mathbf{s}^t$. Hence, if $\mathbf{s}^t$ is a local minimum of $G(\mathbf{s}, \mathbf{s}^t)$ it is also a local minimum of $F(\mathbf{s})$. Thus, by iterating the update of Eq. (3.9) a sequence of estimates is obtained that converges to a local minimum $\mathbf{s}_{min}$ of the objective function $F(\mathbf{s})$:

$$F(\mathbf{s}_{min}) \leq \ldots \leq F(\mathbf{s}^{t+1}) \leq F(\mathbf{s}^t) \leq \ldots \leq F(\mathbf{s}^2) \leq F(\mathbf{s}^1) \leq F(\mathbf{s}^0). \tag{3.10}$$

Accordingly, the convergence of the target function $F = \|\underline{\mathbf{X}} - \underline{\mathbf{AS}}\|$ can now be proven if an appropriate auxiliary function can be determined. This auxilary function is the subject of the following lemma:

Lemma 2. *If* $\underline{\mathbf{K}}(\mathbf{s}^t)$ *is the diagonal matrix*

$$K_{ij}(\mathbf{s}^t) = \delta_{ij}(\underline{\mathbf{A}}^T\underline{\mathbf{A}}\mathbf{s}^t)_i/\mathbf{s}_i^t \tag{3.11}$$

then

$$G(\mathbf{s},\mathbf{s}^t) = F(\mathbf{s}^t) + (\mathbf{s}-\mathbf{s}^t)\nabla F(\mathbf{s}^t) + \frac{1}{2}(\mathbf{s}-\mathbf{s}^t)\underline{\mathbf{K}}(\mathbf{s}^t)(\mathbf{s}-\mathbf{s}^t) \tag{3.12}$$

is an auxiliary function for

$$F(\mathbf{s}) = \frac{1}{2}\sum_i(\mathbf{x}_i - \sum_j \underline{\mathbf{A}}_{ij}\mathbf{s}_j)^2. \tag{3.13}$$

(Here the notation "$\mathbf{x}_i$" denotes the i-th column of the matrix $\underline{\mathbf{X}}$.)

Proof. As can be seen directly in (3.12) $G(\mathbf{s},\mathbf{s}) = F(\mathbf{s})$ holds such that only $G(\mathbf{s},\mathbf{s}^t) \geq F(\mathbf{s})$ needs to be shown explicitly. For this purpose compare the Taylor expansion of $F(\mathbf{s})$

$$F(\mathbf{s}) = F(\mathbf{s}^t) + (\mathbf{s}-\mathbf{s}^t)\nabla F(\mathbf{s}^t) + \frac{1}{2}(\mathbf{s}-\mathbf{s}^t)(\underline{\mathbf{A}}^T\underline{\mathbf{A}})(\mathbf{s}-\mathbf{s}^t) \tag{3.14}$$

with the definition of $G(\mathbf{s},\mathbf{s}')$ in (3.12). Obviously, $G(\mathbf{s},\mathbf{s}^t) \geq F(\mathbf{s})$ only holds if

$$0 \leq (\mathbf{s}-\mathbf{s}^t)[\underline{\mathbf{K}}(\mathbf{s}^t) - \underline{\mathbf{A}}^T\underline{\mathbf{A}}](\mathbf{s}-\mathbf{s}^t). \tag{3.15}$$

In order to see if this inequality can be fulfilled the matrix

$$M_{ij}(\mathbf{s}^t) = \mathbf{s}_i^t(\underline{\mathbf{K}}(\mathbf{s}^t) - \underline{\mathbf{A}}^t\underline{\mathbf{A}})_{ij}\mathbf{s}_j^t, \tag{3.16}$$

which is just a rescaling of $\underline{\mathbf{K}} - \underline{\mathbf{A}}^t\underline{\mathbf{A}}$, is checked for positive semidefiniteness:

$$\begin{aligned}
\mathbf{v}^T\underline{\mathbf{M}}\mathbf{v} &= \sum_{ij} v_i M_{ij} v_j \\
&= \sum_{ij} s_i^t(\underline{\mathbf{A}}^T\underline{\mathbf{A}})_{ij}s_j^t v_i^2 - v_i s_i^t(\underline{\mathbf{A}}^T\underline{\mathbf{A}})_{ij}h_j^t v_j \\
&= \sum_{ij}(\underline{\mathbf{A}}^T\underline{\mathbf{A}})_{ij}s_i^t s_j^t\left[\frac{1}{2}v_i^2 + \frac{1}{2}v_j^2 - v_i v_j\right] \\
&= \frac{1}{2}\sum_{ij}(\underline{\mathbf{A}}^t\underline{\mathbf{A}})_{ij}s_i^t s_j^t(v_a - v_b)^2 \\
&\geq 0.
\end{aligned} \tag{3.17}$$

□

Based on Lemma 1 and Lemma 2 the convergence of Theorem 1 can now be proven as follows:

Proof of Theorem 1: Replacing $G(\mathbf{s}, \mathbf{s}^t)$ in (3.9) by (3.12) results in the update rule

$$\mathbf{s}^{t+1} = \mathbf{s}^t - K(\mathbf{s}^t)^{-1}\nabla F(\mathbf{s}^t). \tag{3.18}$$

Since (3.12) is an auxiliary function, F is nonincreasing under this update rule according to Lemma 1. Finally, if the components in (3.18) are written explicitly the update rule of Theorem 1 is obtained:

$$s_i^{t+1} = s_i^t\frac{(\underline{\mathbf{A}}^T\mathbf{x})_i}{(\underline{\mathbf{A}}^T\underline{\mathbf{A}}\mathbf{s}^t)_i}. \tag{3.19}$$

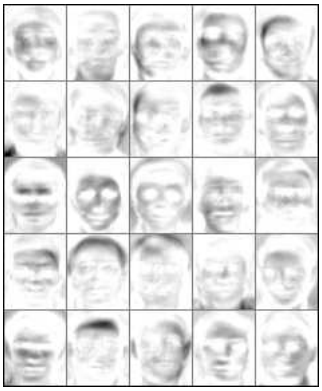

Figure 3.3: Basis images as obtained by NMF when applied to the ORL face image database. Only a global decompostion is obtained. (Figure reproduced from [41])

Analogously, it can be shown that F is nonincreasing under the update rule for $\underline{\mathbf{A}}$ if the roles of $\underline{\mathbf{A}}$ and $\underline{\mathbf{S}}$ are reversed in Lemma 1 and 2

3.2 NMF with sparseness constraints

While NMF lead to the desired sparse decompostion when applied to the face database [1], it only lead to a rather global decompostion when applied to ORL face image database [2] (see Fig. 3.3). For this reason, an extension of the standard NMF procedure is presented in [41] in which the sparseness of the matrices $\underline{\mathbf{A}}$ and $\underline{\mathbf{S}}$ can be enforced by an additional projection step (see below).

In this algorithm, the sparseness σ of an N-dimensinal vector $\mathbf{x}$ is defined by means of its L_1 and L_2 norm:

$$\sigma(\mathbf{x}) = \frac{\sqrt{N} - (\sum_{i=1}^{N} |x_i|) / \sqrt{\sum_{i=1}^{N} x_i^2}}{\sqrt{N} - 1}. \tag{3.20}$$

Although this measure does not count the nil entries of $\mathbf{x}$ it is still a good approximation for its sparseness as

1. $\sigma(\mathbf{x}) = 1$ if and only if $\mathbf{x}$ contains only a single non-zero component,
2. $\sigma(\mathbf{x}) = 0$ if and only if all components are equal,
3. $\sigma(\mathbf{x})$ interpolates smoothly between these two extremes.

The last point is especially appealing as a sparseness measure like

$$\sigma_0(\mathbf{x}) = \frac{\#\text{ of nil entries of } \mathbf{x}}{N} \tag{3.21}$$

that directly counts the nil entries of $\mathbf{x}$ is not continous and hence hard to handle in algorithms.

In the algorithm proposed in [41] a factorization as in (3.2) is now sought such that the columns of $\underline{\mathbf{A}}$ as well as the rows of $\underline{\mathbf{S}}$ have predefined sparsenesses $\sigma_{\underline{\mathbf{A}}}$ and $\sigma_{\underline{\mathbf{S}}}$ respectively:

Definition 2. *NMF with sparseness constraints*
Given a non-negative matrix $\underline{\mathbf{X}}$ of size $m \times T$, find two nonnegative matrices $\underline{\mathbf{A}}$ and $\underline{\mathbf{S}}$ of sizes $M \times N$ and $N \times T$ respectively such that

$$F(\underline{\mathbf{A}}, \underline{\mathbf{S}}) = \|\underline{\mathbf{X}} - \underline{\mathbf{A}}\ \underline{\mathbf{S}}\|^2 \tag{3.22}$$

is minimized under the (optional) constraints

$$\sigma(\mathbf{a}_i) = \sigma_{\underline{\mathbf{A}}} \tag{3.23}$$

$$\sigma(\mathbf{s}_i) = \sigma_{\underline{\mathbf{S}}} \tag{3.24}$$

htb

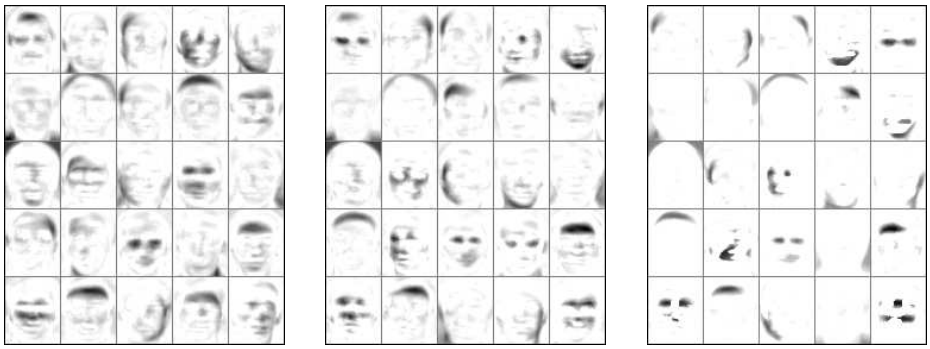

Figure 3.4: Features learned from ORL face image database using NMF with sparseness constraints. If the sparseness of the basis images is increased the representation switches from a global one to a local one. Sparseness leves were set to 0.5 (left), 0.6 (middle), 0.7 (right) (figure reproduced from [41]).

where $\mathbf{a}_i$ *is the* i*-th* column *of* $\underline{\mathbf{A}}$ *and* $\mathbf{s}_i$ *is the* i*-th* row *of* $\underline{\mathbf{S}}$*. Here,* N *denotes the desired number of components and* $\sigma_{\underline{\mathbf{A}}}$ *and* $\sigma_{\underline{\mathbf{S}}}$ *are the desired sparseness values. These three paramters are provided by the user.*

Note, that in (3.20) the L_1 and L_2 norm of the vector $\mathbf{x}$ are used to determine its sparseness. Because of the scaling indeterminacy inherent in the BSS problem one of these norms can be fixed for either the columns of $\underline{\mathbf{A}}$ or the rows of $\underline{\mathbf{S}}$. Hence, in the sequel the L_2 norm of the rows of $\underline{\mathbf{S}}$ will be set to unity for convenience.

3.2.1 Algorithm for NMF with sparseness constraints

The algorithm proposed in [41] primarily consists of the two following steps. First, either the matrix $\underline{\mathbf{A}}$ or $\underline{\mathbf{S}}$ is updated by gradient descend such that the (3.22) is minimized. Second, a projection operator is used which enforces the desired sparsenesses of $\underline{\mathbf{A}}$ and $\underline{\mathbf{S}}$ respectively. This projection operator will

be discussed in detail after the outline of the algorithm.

Input: observation data matrix $\underline{\mathbf{X}}$, sparseness constraints $\sigma_{\underline{\mathbf{A}}}$, $\sigma_{\underline{\mathbf{S}}}$

Output: decomposition $\underline{\mathbf{A}}$ $\underline{\mathbf{S}}$ of $\underline{\mathbf{X}}$ such that rows of $\underline{\mathbf{S}}$ fulfill given sparseness constraints $\sigma_{\underline{\mathbf{A}}}$ and such that the columns of $\underline{\mathbf{S}}$ fulfill $\sigma_{\underline{\mathbf{S}}}$

1 Initialize $\underline{\mathbf{A}}$ and $\underline{\mathbf{S}}$ to a random non-negative matrix.
2 Project each column of $\underline{\mathbf{A}}$ to be nonnegative, have unchanged L_2 norm but L_1 norm set to achieve desired sparseness $\sigma_{\underline{\mathbf{A}}}$.
3 Project each row of $\underline{\mathbf{S}}$ to be be nonnegative, have unit L_2 norm and L_1 norm set to achieve desired sparseness $\sigma_{\underline{\mathbf{S}}}$.
repeat
4 Set $\underline{\mathbf{A}} \leftarrow \underline{\mathbf{A}} - \mu_{\underline{\mathbf{A}}}(\underline{\mathbf{A}}\ \underline{\mathbf{S}} - \underline{\mathbf{X}})\underline{\mathbf{A}}^T]$.
5 Project each column of $\underline{\mathbf{A}}$ to be nonnegative, have unchanged L_2 norm, but L_1 norm set to achieve desired sparseness $\sigma_{\underline{\mathbf{A}}}$.
6 Set $\underline{\mathbf{S}} \leftarrow \underline{\mathbf{S}} - \mu_{\underline{\mathbf{S}}}\underline{\mathbf{A}}^T(\underline{\mathbf{A}}\ \underline{\mathbf{S}} - \underline{\mathbf{X}})$
7 Project each column of $\underline{\mathbf{S}}$ to be nonnegative, have unit L_2 norm, but L_1 norm set to achieve desired sparseness $\sigma_{\underline{\mathbf{S}}}$.
until *convergence*;

In the above algorithm gradient descent is used to update $\underline{\mathbf{A}}$ and $\underline{\mathbf{S}}$ first. Note however, that if either no sparseness constraints for either $\underline{\mathbf{A}}$ or $\underline{\mathbf{S}}$ are provided by the user the multiplicative update rules (3.5) may be used.

With this algorithm, the desired parts-based decomposition of the face images in [2] could be achieved. Fig. 3.4 shows the result for different sparseness values.

Obviously, a central part of this algorithm is the projection operator applied to ensure that the rows of $\underline{\mathbf{S}}$ as well as the columns of $\underline{\mathbf{A}}$ have the desired sparseness. While this projection operator was proposed heuristically by Hoyer in [41] we could proof its uniqueness in [86] in the context of the *extended sparse NMF algorithm* which will be presented in Sec. 7.1. Hence,

this operator will not be discussed here but also in Sec. 7.1.

Chapter 4

Genetics

Genetics is one of the most active and growing research fields of modern biology. In recent years, it lead to striking results like e.g. the decoding of the human genome or the cloning of "Dolly", a sheep which was the first mammal to be reproduced from an adult cell. Apart from these spectacular results genetic research also lead to very practical applications like, for instance, the production of insulin by bacteria but also helped to elucidate the causes of many hereditary diseases.

In this chapter, the most important principles of molecular genetics will be presented. Based on the Central Dogma of Molecular Biology, which states how hereditary information is used to synthesize proteins, the term "gene" will be defined precisely. Furthermore, the so-called transcription process of genes as well as its regulation by transcription factors will be elucidated for both prokaryotic and eukaryotic cells. Finally, the synthesis process of proteins, which are the end product of any genetic process, will be illuminated.

Note, that for the reader not familiar with the basic structure of biological cells or the genetic code a short review of these concepts is presented in Appendix A.

4.1 Gene expression

4.1.1 Central Dogma of Molecular Biology and Genes

Today, it is a well known fact that all the information about an organism is encoded into its DNA (see A for details). Before a cell can benefit from this information, however, an intermediary step called transcription is necessary. Thereby, a copy of a coding segment of DNA is made in the form of an RNA molecule. This molecule is sent to one of the ribosomes where its information is used to synthesize a protein in a process called translation.

This flow of information

$$\text{DNA} \xrightarrow{\text{Transcription}} \text{RNA} \xrightarrow{\text{Translation}} \text{Protein} \tag{4.1}$$

is known as the *Central Dogma of Molecular Biology*.

This dogma is also used to define the term "gene": regions of DNA that transmit information to RNA are called genes, or, in other words, a gene is defined by the *initial* RNA molecule that is transcribed from it. As will be seen later, the second definition is necessary for eukaryotic cells where the initial RNA molecule undergoes some postprocessing before its information is used.

Both of the essential processes appearing in the Central Dogma, transcrip-

tion and translation, will be explained in detail in the following sections.

4.1.2 Transcription

In this section, the molecular mechanisms of the transcription process will be elucidated in detail. The basic concepts of these mechanisms are the same in prokaryotic and eukaryotic cells, whereas, however, eukaryotic transcription is in many details much more complicated than its prokaryotic counterpart. Hence, prokaryotic transcription will be explained first in the following while the main additional features of eukaryotic transcription will be discussed afterwards.

4.1.2.1 Transcription in Prokaryotes

As already explained in 4.1.1 copies of the genes found in DNA have to be made before their information can be used to synthesize specific proteins. This copying process is carried out with the aid of an enzyme called RNA polymerase. In order to get recognized by RNA polymerase, genes must distinguish themselves from other, noncoding parts of the DNA. In both prokaryotic and eukaryotic cells the distinctive feature is a certain segment of code, called promoter, which is usually found directly adjacent to the gene in the upstream direction. In prokaryotic cells the promotors generally contain two typical DNA sequences. The first is found at the -10 position (i.e. ten basepairs upstream of the beginning of the gene) and contains the sequence TATAAT while the second contains the sequence TTGACA and is located at the -35 position.

Prokaryotic RNA polymerase is capable of recognizing these promotor regions. Functionally, it consists of two major parts, a sigma subunit and a so-called core enzyme built of two alpha, one beta and one beta prime subunit. This configuration is called the RNA polymerase holoenzyme. While the core enzyme is responsible for the polymerase activity the sigma subunit is used to detect promoter regions in the DNA. Once the sigma-subunit recognizes a TTGACA sequence it binds the core enzyme to the DNA and separates then from the polymerase complex. Under the influence of the core enzyme the DNA unwinds at the TATAAT sequence and forms a so-called transcription bubble, a small region of the DNA (about 18 base pairs in length) where the two DNA strands are separated and unwinded. Collectively, these steps are summarised as the initiation of the transcription procress.

In the next step, usually referred to as elongation, the RNA molecule is synthesized by the core enzyme. For this purpose the 3' to 5' strand of the transcription bubble is used as a template. At this template strand base pairing between its A, T, C, and G bases and the complementary ribonucleotids UTP, ATP, GTP and CTP, which are usually found in abundance in the cell, occurs. Next, an RNA chain is built by the core enzyme which links the single RNA molecules attached to the template strand by phosphordiester bonds. After these bondings have been established the core enzyme proceeds in the 3' to 5' direction of the template strand. In front of the core enzyme a new transcription bubble is formed while behind the core enzyme the RNA-DNA hybrid duplex quickly dissociates and the DNA rewinds. Hence, always only a small number of RNA nucleotides is bound to the DNA strand. As a result, the transcription bubble appears to move along the gene with the

polymerase while behind the core enzyme an RNA chain seems to grow.

In the last step of the transcription process, called termination, the transcription complex is separated from the DNA strand. In prokaryotic cells, two termination mechanisms are known. The first, called rho-dependent termination, is carried out with the aid of a protein called rho and requires that at the end of the gene a region rich in C is found. During the transcription process rho attaches to the 5' end of the growing RNA molecule and moves along the RNA towards the core enzyme. When the core enzyme arrives at the C rich segment of the gene it pauses such that rho can catch up with it. Once arrived at the core enzyme rho knocks the polymerase as well as the RNA off the DNA and terminates the transcription process.

In the second termination mechanism, called rho-independent termination, no additional protein is needed. However, a G/C rich segment of nucleotides followed by an A/T rich block has to appear at the end of the gene. After the G/C rich segment got transcribed by the core enzyme it forms a short double-stranded region called hairpin. This hairpin slows the core enzyme down, causing it to pause at the following A/T rich region. As the A/T pairs are comparatively weak the transcription complex finally falls apart such that the RNA chain fully separates from the DNA and the transcription process is finished.

4.1.2.2 Transcription in Eukaryotes

Even if basically similar eukaryotic transcription differs in many aspects from its prokaryotic counterpart. For instance, only one type of RNA polymerase is used for the transcription of all types of RNA in prokaryotic cells while

in eukaryotic cells (at least) three different types of RNA polymerase are needed. Thereby, each type of RNA polymerase synthesizes exactly one type of RNA as outlined in the following table:

RNA polymerase I	Genes encoding ribosomal RNA
RNA polymerase II	Genes encoding messenger RNA
RNA polymerase III	Genes encoding transfer RNA

In the following, the transcription process in eukaryotic cells will be discussed considering as example the synthesis of mRNA by RNA polymerase II (RNAP II).

Like in prokaryotic genes promoters are usually found upstream and directly attached to eukaryotic genes. They have the same function as in prokaryotic DNA, though, the transcription of eukaryotic genes is additionally steered by further regulatory DNA sequences. These DNA regions may lay upstream, downstream or even in their related gene and can be used to adjust its transcription rate.

As there are many regulatory sequences associated with eukaryotic genes, the term promoter needed to be further precised in order to avoid ambiguities. This has been achieved by introducing the term "core promoter" in the context of eukaryotic cells. It is defined as the minimal stretch of continuous DNA sequence that is sufficient to direct accurate initiation of transcription by the RNA polymerase II machinery [16]. Typically, the core promoter encompasses the site of transcription initiation and extends either upstream or downstream for an additional 35 basepairs. Four types of special DNA sequences can be found in core promoters: a TATA-box, a TFIIB recognition

element (BRE), an initiator (Inr) and a downstream core promoter element (DPE). It must be noted, however, that not all of these four elements can be found jointly in all core promoters.

In contrast to prokaryotic RNA polymerase, RNAP II can only bind to core promoters by means of five auxiliary proteins called general transcription factor (TF) IID, TFIIB, TFIIF, TFIIE, and TFIIH. The first to bind to the core promoter is TFIID which functionally consists of a TATA-binding protein (TBP) as well as about 13 TBP associated factors (TAFs) [16]. TBP is used by TFIID to detect and attach itself to promoters containing a TATA-box. TATA-free promoters can also be detected by TFIID with the aid of its TAFs. These proteins are able to bind sequence specific to the Inr and DPE region of the core promoter and are furthermore used to mediate remote activation signals of the gene [18] (see 4.1.3.2). After TFIID is bound TFIIB can recognise the BRE of the core promoter to which it eventually binds to. Jointly, TFIID and TFIIB are now capable of recruiting RNAP II together with TFIIE to DNA whereas TFIIE is used to speed up the polymerisation process. Next, TFIIE and TFIIH enter the complex. They are needed to allow RNAP II to move along the DNA molecule during the transcription process, to unwind the DNA helix and to determine which of the two DNA strands will be transcribed.

Finally, a complex called Mediator binds to the TFII-RNAP II complex. Mediator, which was discovered in yeast in 1990 and in mammals in 1998, was originally only assumed to mediate signals from remote regulatory regions to the core promoter. Recent research [87] has shown, however, that the presence of mediator at the core promoter is indispensable for the initiation

of the transcription process *in vitro* and *in vivo*.

Once mediator, the TFIIs and RNAP II have bound to the core promoter the corresponding gene is transcribed at a so-called basal transcription rate. This rate can be altered deliberately via further transcription factors which bind to the remote regulatory regions of the gene. Depending on if occupancy by their cognate transcription factors leads to increasing or decreasing transcription rates, these remote regulatory regions are called enhancers or repressors, respectively. The mechanisms for adjusting the transcription rate by additional factors will be discussed in detail in 4.1.3.2.

While the initiation process differs significantly between eukaryotic and prokaryotic cells no differences are found during the elongation process. However, further differences appear during the termination process as eukaryotic genes do not exhibit any termination sequences which are recognised by RNAP II. Hence, RNAP II has no indication about when to stop the transcription process and often transcribes up to 2000 additional nucleotides downstream of the actual region of interest.

Even if not recognised by RNAP II, the position of the actual 3' end is still determined by a sequence within the mRNA itself. This sequence, AAUAAA, is known as the polyadenylation signal and is recognised by certain enzymes. These enzymes start to cleave the mRNA 10 to 30 nucleotides downstream of the signal, and add a series of about 100 to 250 adenine nucleotides. This process is called polyadenylation and is carried out without a template, i.e. the As are simply added one after another to the 3' end of the mRNA. The length of this polyadenine tail determines the lifespan of the produced mRNA molecule. The longer the tail is, the more stable and longer-lasting is the

mRNA.

Apart from polyadenylation a capping of the mRNA occurs at the 5' end. Thereby, a methylated guanine nucleotide is added to the mRNA molecule in a 5' to 5' phosphordiester bond. This cap is needed by the mRNA in order to be recognised by the ribosomes in the further course of gene expression.

Finally, there is another remarkable difference between prokaryotic and eukaryotic mRNA. In eukaryotic cells, parts of the initial mRNA molecule (called precursor mRNA or pre-mRNA) are removed in a process called splicing shortly after transcription is finished. The segments to be excised from the molecule are called introns while the remaining parts are called exons. Introns can be found anywhere in the pre-mRNA molecule and often lie in between exons. Many introns start from the sequence GU and end with AG (in 5' to 3' direction) whereas the first sequence is referred to as splice donor and the second as splice acceptor. About 20 to 50 base pairs upstream of the acceptor another typical sequence, CU(A/G)A(C/U) (here (base1/base2) means that either base1 or base2 is found in the sequence), called branch site, is found in introns. Between this branch site and the AG end of the intron a segment rich in C or U, called polypyrimidine sequence, is also typical. Furthermore, about 60 % of all exons have the sequence (A/C)AG abutting to the donor site of the intron as well as a G base at the acceptor site (see Fig. 4.1 A).

In some pre-mRNAs introns can detach themselves from the molecule without the aid of additional enzymes, i.e. the enzymatic activity resides within the intron pre-mRNA itself. More often, though, the intron is removed by means of five small nuclear ribonucleoproteins (snRNPs) called U1, U2,

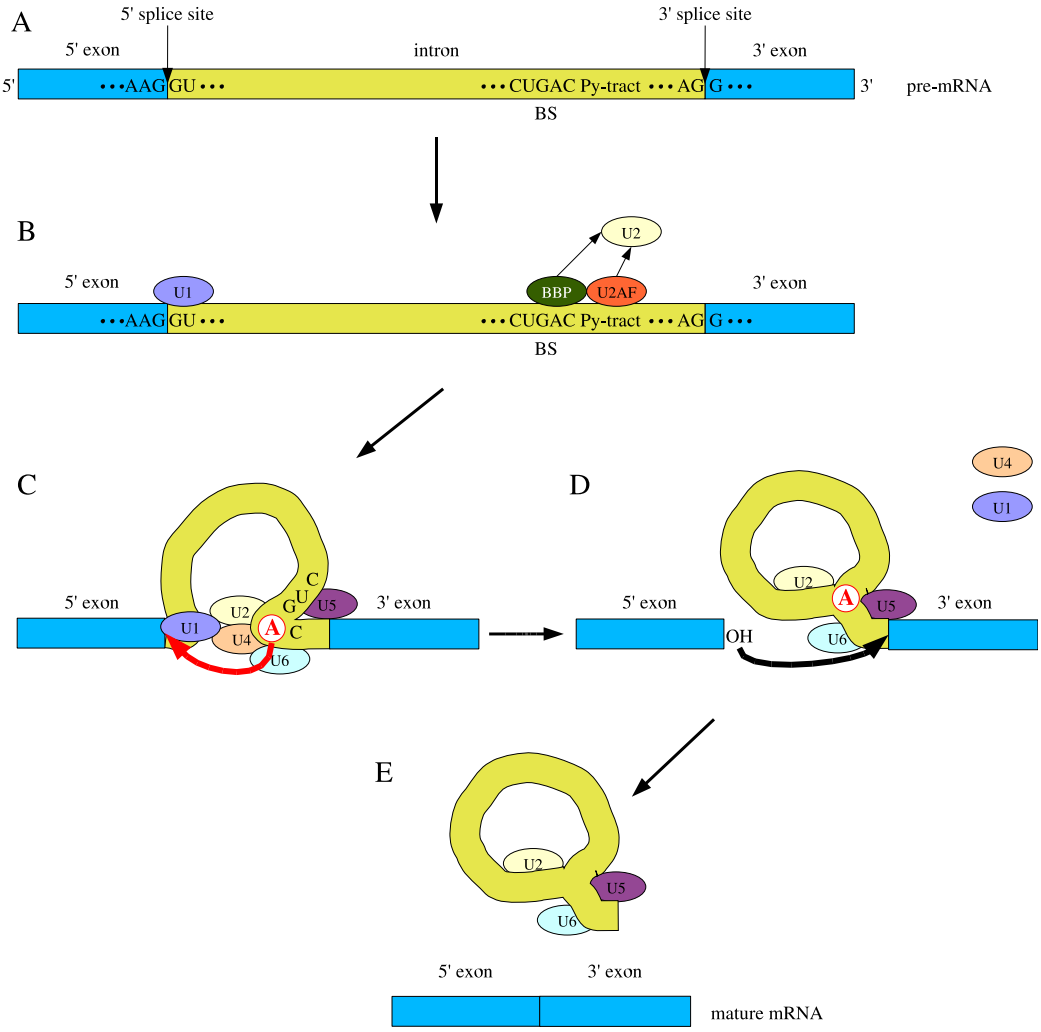

Figure 4.1: Splicing process. A) pre-mRNA containing one intron. Typical sequences at exon/intron boundaries as well as the branch site (BS) and the polypyrimidine segment (Py-tract) are shown. B) While the snRNP U1 is capable of detecting and binding to the 5' splice site by itself, two additional proteins, namely branchpoint bridging protein (BBP) and U2 auxiliary factor (U2AF) are needed to attract U2 to mRNA. Thereby, BBP recognizes and binds to the branch site while U2AF attaches itself to the polypyrimidine sequence. C) After U1 and U2 also U4, U5 and U6 bind to mRNA, collectively building a spliceosome (BBP and U2AF not shown for simplicity). This spliceosome leads to a curving of the intron sequence upon which the A nucleotide (shown in red) of the BS attacks the 5' splice site and cleaves it. D) The cut 5' end of the intron sequence becomes covalently linked to the BS's A nucleotide and U1 and U4 leave the spliceosome. Furthermore, the 3'-OH end of the first exon sequence, which was created in the first cleavage, adds to the beginning of the second exon sequence, cleaving the mRNA molecule at the 3' splice site. Thus, the intron together with the snRNPs is eventually separated from mRNA and the mature mRNA molecule is produced (E).

U4, U5 and U6 as well as at least two additional proteins termed branchpoint bridging protein (BBP) and U2 auxiliary factor (U2AF). At the beginning of the intron cleavage process U1 detects and binds to the GU sequence at the 5' end of the intron. At the same time BBP attaches to the branch site and U2AF occupies the polypyrimidine segment close to the intron's 3' end. Together, BBP and U2AF attract U2 to the branch site where it binds to. Upon U1 and U2 binding the remaining snRNPs U4, U5 and U6 join the snRNP-mRNA complex thus building a so-called spliceosome. This spliceosome then cleaves the intron first at its 5' and then at its the 3' end and eventually joins adjacent exons together to produce the mature mRNA molecule (see Fig. 4.1 for details). As will be discussed in 4.1.3.2 the splicing process is often not unique meaning that a single type of pre-mRNA can lead to a multitude of different mature mRNA molecules. In such cases one often speaks of alternative splicing.

4.1.3 Gene regulation

The last section dealt with how genes are detected and transcribed by RNA polymerase. However, not all the genes found in the DNA need to be transcribed all the time. For instance, some cells are known to react to high environmental temperatures by protrucing so-called heat shock proteins. These proteins help the cell to endure heat stress but are useless at moderate temperatures. Thus, it would be an enormous waste of energy to synthesize these proteins permanently even if the ambient temperatures are low. Accordingly, cells only transcribe the genes coding for those heat shock proteins

at high temperatures. This means, that there must be a regulatory mechanism telling RNA polymerase which genes are needed under some given circumstances and which ones should be ignored. This mechanism is elucidated in more detail in this section.

4.1.3.1 Gene Regulation in Prokaryotic Cells

In prokaryotic cells genes that affect the same biochemical pathway (e.g. constitution of heat shock proteins) are transcribed under the same conditions. Such genes can be found as consecutive segments along the DNA. Their transcription is jointly steered by only one regulatory unit consisting of a promotor as well as a so-called operator. This operator is a segment of DNA located between the promoter and the genes. Collectively, the genes together with their promotor and their operator are referred to as an operon. To the operator a specific protein called repressors can attach. If no repressors are present RNA polymerase can bind to the promotor, advance to the genes and transcribe them. If a repressor has bound to the operator, however, RNA polymerase is kept from advancing to the genes such that no transcription can occur. Hence, the question when a specific group of genes is transcribed is boiling down to the question under which circumstances a suitable repressor exists or not.

To answer this question two different types of operons have to be distinguished. In the first case of so-called inducible operons, the repressors are always produced by the cell and are capable of attaching themselves to the operator regions. Thus, they would block the transcription process permanently. However, the repressors lose their inhibiting qualities when certain,

repressor specific molecules appear in the cell. These molecules bind to the repressors and change their conformation such that they cannot bind to the operators anymore. Accordingly, the transcription process of the genes in the operon starts. A famous example for this mechanism can be found in the lactose metabolism. If no lactose is present in the cell the so-called lac repressor attaches to the operator regulating the lactose operon and keeps any genes from being transcribed. If lactose enters the cell the lac repressor binds to it and changes its conformation. Thus, it cannot attach to the lactose operator anymore and the genes necessary for the lactose metabolism are transcribed. In other words, the lactose metabolism is simply induced by the presence of lactose in the cell.

In the second type of operons, called repressible operons, the repressor is again produced permanently. However, the mere repressor is not yet capable of attaching itself to its corresponding operator. Hence, the genes in the operon are transcribed and the corresponding proteins are produced. Usually, the repressor then binds to the end product of the protein synthesis process and changes its conformation such that it can attach to the operator. Thus, the process stops as soon as its end product is built.

4.1.3.2 Gene Regulation in Eukaryotic Cells

In prokaryotic cells gene regulation is mostly used to respond to changing ambient conditions and to steer the cell reproduction. In eukaryotic organisms, however, there is yet another important reason for gene regulation. Eukaryotic organisms usually consist of hundreds of different kinds of cells whereas each cell type is specialised to fulfil a specific function in the or-

ganism. Which kind of function a specific cell type may have depends on the genes it expresses. Hence, the gene regulatory mechanism in eukaryotic cells not only needs to account for the environmental conditions and the cell reproduction, but must also be capable of steering the gene transcription process cell dependently.

Obviously, such a gene regulatory mechanism needs to be more sophisticated than in prokaryotes. Accordingly, genes are no longer organised into operons in eukaryotes but are regulated individually. Technically, the regulatory mechanism is based on the interactions between the additional (nonpromoter) regulatory domains of each gene and a set of nongeneral transcription factors. Depending on their type, current state and environmental conditions cells produce or activate certain sets of these transcription factors whereas the composition of the sets codes for what kinds of genes are currently needed by the cell. Eukaryotic genes are able of sensing these transcription factors via their additional regulatory domains whereas to each of these domains only certain transcription factors can bind sequence specifically. Depending on which transcription factors are bound the corresponding gene is eventually expressed at a certain rate.

Furthermore, as already adumbrated in 4.1.2.2, the initial mRNA molecule in eukaryotes is often subject to an important postprocessing mechanism called alternative splicing. Thereby, depending on cell type and environmental conditions, different sets of introns are cleaved from pre-mRNA which means that a single gene can give rise to a multitude of different types of mature mRNA. Again, this process is steered by nongeneral transcription factors.

In the following each of the above steps of eukaryotic gene regulation will be elucidated in detail.

Nongeneral Transcription Factors [14] The nongeneral transcription factors mentioned above are modular proteins bearing distinct regions which are dedicated to different functions: a DNA binding domain used to detect and bind to specific DNA sequences, a multimerisation domain allowing the assembly of either homo- or hetero-multimers (compositions of several proteins of the same or of various types) as well as an effector domain that can modulate (activate or repress) gene transcription rates. While the DNA binding domain is an inherent part, multimerisation domains do not appear in all nongeneral transcription factors. Furthermore, the effector domains often need further proteins, called cofactors, to function properly.

Basically, the nongeneral transcription factors can be divided into constitutively active nuclear factors and regulatory transcription factors.

The first ones can be found permanently in the nuclei of all cells and are known to have transcriptional activation potential *in vitro*. However, these factors by themselves do not change the expression rate of individual genes in a chromosomal context. They are rather assumed to have an important facilitating role in the transcription of genes which are permanently needed, like, e.g. genes coding for structural proteins or for ubiquitous metabolic enzymes. Furthermore, their presence is often indispensable for other transcription factors to work properly.

Regulatory transcription factors, in contrast, are capable of changing the transcription rates of individual genes *in vivo* and are needed for cell differ-

entiation as well as for coping with changing environmental conditions. In [14] they are further divided into two major subgroups as follows:

Developmental transcription factors. These factors are produced cell specifically starting from the beginning of the cell as a fertilezed egg. They are expressed in sequential waves during the various stages of cellular development. Even if their regulatory mechanism is not fully understood yet they mostly seem to be steered by external signals. Once being produced they enter the nucleus immediately without requiring any regulated posttranslational event. In the nucleus they lead to cell specific expression of proteins, i.e. they are responsible for cell differentiation.

Signal dependent transcription factors. These proteins are expressed in most or all cells of an organism but are inoperative until they receive a specific intra- or extracellular activation signal. Three major groups of signal dependent transcription factors are distinguished.

- Steroid receptors. These transcription factors reside in the nucleus of the cell and are activated by external signals in the from of steroids [59]. These steroids may enter the cell on their own and induce an allosteric change when bound to their cognate receptors. This change allows the steroid-receptor complex to bind to high affinity sites in chromatin and modulate transcription. In humans about 50 of these steroid receptors are known.

- Transcription factors activated by cell surface receptor-ligand interactions. [14] The majority of these transcription factors resides in the nucleus of the cell and may even be bound to the DNA molecule.

Their activation process is initiated when extracellular signalling proteins bind to their corresponding receptors at the cell surface. This binding eventually leads, via some intermediate steps, to a phosphorilation of the serine nucleoacids of these transcription factors. Upon this phosphorilation the transcription factors get activated and steer the transcription of other genes. It may be noted that there are many serine kinases which may lead to the phosphorilation of the same transcription factor. Additionally, the number of this kind of transcription factors seems to be very high. Hence, it has been difficult to trace precise protein pathways to the activation of individual genes in a natural context.

Apart from those residing in the nucleus a smaller number of inactive transcription factors can also be found in the cytoplasm. Their individual activation mechanisms vary widely (for a review see e.g. [59]) and a detailed elucidation of them is far beyond the scope of this thesis. In brief, some of these factors are activated directly at the cell surface receptor by serine or tyrosine phosphorilation. Others are activated in the cytoplasm via cytoplasmic serine phosphorilation and/or proteolysis. Finally, there are also transcription factors known which are activated via fluctuations in so-called second messengers like e.g. Ca^{2+} or NFAT (nuclear factors in activated T cells). All of these transcription factors enter the nucleus after activation and participate at regulatory processes of genes.

- Transcription factors activated by internal (cell-autonomous) signals.

Recent research has revealed a new class of transcription factors which are activated by intracellular signals [15]. The precursors of these transcription factors can be found in some transmembrane proteins which are located in the cell membrane or in the membrane of the endoplasmatic reticulum. One feature of these proteins is that they span the membranes always in the same direction. From these proteins the transcription factors are cleaved in a process called regulated intramembrane proteolysis (Rip). Rip is triggered by certain intracellular signals like, e.g. the internal sterol concentration or the presence of unfolded proteins. These signals are sensed specifically by other proteins which then cleave the protein hosting the corresponding transcription factor at its extracytosolic (i.e. extracellular or residing in the lumen of the endoplasmatic reticulum) face. Once the host protein is cleaved extracytosolically it attracts further proteins which carry out a second intramembrane cleavage. In this process, the transcription factor is cut off its host protein whereon it is heading for the nucleus. Once arrived there it usually steers the transcription of those genes which are associated with its own activation signal.

Finally, it must be mentioned that some of the proteins which are expressed after their corresponding gene was activated by the transcription factors discussed so far, may act themselves as transcription factors for other genes. Often, this second transcription factor as well as the original transcription factor are then needed simultaneously to steer the expression of the eventual target gene.

Transcription regulation by nongeneral transcription factors As already described in 4.1.2 the transcription process of eukaryotic genes is initiated as soon as the basal transcription machinery, consisting of RNAP II, TFIIs, TAFs and mediator, binds to the core promoter. Most core promoters, however, are to weak to attract the basal transcription machinery by themselves, i.e. their genes are not transcribed (or only at very low levels) autonomously. In these cases the transcription process can only be initiated by means of the nongeneral transcription factors discussed above. Once having received their specific activation signals these factors use their DNA binding domains to detect and attach themselves sequence specifically to the enhancer regions of the genes they control. After a transcription factor has bound to its cognate enhancer its effector domain starts, in the simplest case, to interact directly with the TAFs or the mediator complex of the basal transcription machinery. These interactions lead to the build-up of the basal transcription complex at the core promoter of the corresponding gene which is then transcribed.

In higher eukaryotic cells, however, cases in which a single transcription factor suffices to initiate the transcription process are very rare. Usually, the enhancer region of a gene consists of several binding sites to which a certain, gene specific set of transcription factors can bind to. Only if all these transcription factors have bound to the enhancer they can cooperatively start to recruit the basal transcription machinery to the core promoter. Often this can only be achieved by means of additional proteins called coactivators.

A well studied example of cooperativity between several transcription factors and coactivators can be found in the regulation of interferon-β (IFN-

β) (reviewed in [62]), a gene which is involved in the regulation of immune responses against viral infections. The expression of IFN-β is steered by a TATA-box containing core promoter as well as by an enhancer. *In vivo*, the core promoter sequence is bound to a nucleosome making it invisible to the basal transcription machinery. Hence, the gene is normally kept in an 'off' state. The enhancer, in contrast, is nucleosome free and is situated next to the nucleosome containing the core promoter. To it four typical nongeneral transcription factors, namely nuclear factor κB (NF-κB), interferon regulatory factor (IRF) 3 and 7, activating transcription factor 2 (ATF-2), as well as an architectural factor called high mobility group protein I(Y) (HMG I(Y)) can bind sequence specifically. The latter attaches itself to four sites of the enhancer [97] leading to an unbending of the DNA molecule and facilitating the binding of NF-κB, IRF3/7 and ATF-2. After these typical transcription factors have bound to the enhancer, HMG I(Y) furthermore orchestrates a network of protein-protein interactions between them as well as between itself and them. These interactions lead to a remarkably stable nucleoprotein complex termed the IFN-β enhanceosome.

To this enhanceosome a complex called GCN5 is recruited which acetylates, among others, the nucleosome hosting the core promoter of IFN-β. Upon this acetylation, a coactivator called CBP binds to the enhanceosome and mediates the recruitment of SWI/SNF as well as the basal transcription machinery to the core promoter. SWI/SNF remodels the IFN-β nucleosome such that the core promoter is exposed and finally bound to by the basal transcription machinery. IFN-β then gets transcribed until HMG I(Y) undergoes a second acetylation by the coactivator CBP. This acetylation leads to the

disassembly of the enhanceosome and thus to the end of the transcription process.

In contrast to IFN-β there are also genes whose core promoters are strong enough to recruit the basal transcription machinery by themselves, i.e., these genes are transcribed automatically without activation by any nongeneral transcription factors. These genes can be silenced by mechanisms similar to those found in the regulation of IFN-β (reviewed in [27]). Again, multiple gene specific transcription factors have to bind simultaneously to the non-promoter regulatory region (but now to the repressor) of the gene they control. These transcription factors then interact with each other and build a so-called repressosome, i.e. a nucleoprotein complex similar to the enhanceosome as found in IFN-β. Often, a corepressor (like e.g. Groucho) binds to this repressosome and silences the corresponding gene. Thereby, two mechanisms are found frequently. On the one hand, many corepressors functionally interact with histone deacetylases. These deacetylases then modify the chromatin structure such that the core promoter of the related gene is hidden from the basal transcription machinery. On the other hand, some corepressors compete with the core promoter for parts of the mediator complex. In that case, parts of the mediator complex bind to the corepressor and are therefor no longer at the disposal of the basal transcription machinery. Thus, the transcription process is inhibited.

So far, the discussed mechanisms targeted only at regulating specific genes. There are, though, a number of general repressors which interfere with the transcription process of all genes in a cell (for a review see [58]). These general repressors are used, for instance, when cells sense low levels

of nutrition and hence reduce their overall activity. Most of these general repressors function by hampering the constitution of the basal transcription machinery at the core promoter. A good example of a general repressor is NC2 which attaches to TFIID during the build-up of the basal transcription machinery. Upon this binding interactions between TFIIB and TFIID are inhibited such that the formation of the basal transcription machinery is blocked. Other general repressors remove the TBP from the core promoter (e.g. Mot1 in yeast) or alter RNAP II such that it cannot form a complex with the general transcription factors (e.g. Srb10/Srb11 complex in yeast, Cdk8/cyclin C in mammalian cells).

Transcription regulation by multiple regulatory regions Until now, the considerations and the examples only dealt with genes which were controlled by a single enhancer or repressor region. In higher, multicellular eukaryotes, however, many genes are steered by large regulatory elements consisting of several enhancer or repressor regions. This increased regulatory complexity became necessary in the course of evolution as the number of factors determining when and at which rate a gene should be transcribed increased with the intricacy of the organisms.

The interactions between the individual enhancers and repressors steering a single gene may become quite complicated as was shown in an investigation of Endo16 [99], a gene that encodes a polyfunctional secreted protein of the midgut in the late embryo end larva of sea urchins [85].

The transcription of this gene is controlled by a 2300 base pairs long regulatory element abutting to the core promoter in the upstream direction.

It contains two enhancers and three repressors as well as a multifunctional regulatory unit which lies closest to the core promoter and functions either as an enhancer or as a signal transducer. When being the only activated (i.e. bound to by its cognate transcription factors) regulatory region of Endo16, this multifunctional element serves an enhancer and leads to average transcription rates. If one or both of Endo16's regular enhancers are activated additionally they interact with the multifunctional element and synergetically increase the transcription rate significantly. In contrast, if the multifunction element as well as any of the repressors are activated simultaneously the transcription process is inhibited, i.e. the activation of any of Endo16's regular enhancers is then ignored entirely. However, if the multifunctional element is inactive any activation of the repressors is ignored and transcription of Endo16 can always be initiated by activation of its regular enhancers. Hence, the multifunctional regulatory element is obviously necessary to mediate the signals from the repressors to the core promoter.

Gene Regulation by Alternative Splicing When the human genome was decoded at the beginning of this decade it was a great surprise that a complex organism like the human being has all its hereditary information stored in only 20000 to 30000 genes. Given the "one gene codes for one protein" hypothesis that prevailed widely until the last decade this number seemed to be far too low to generate all the proteins needed in humans. This discrepancy became clarified with the progressive understanding of the role of alternative splicing, a process which was first discovered in 1980 [56] but which had been considered as an exception for a long time. Recent

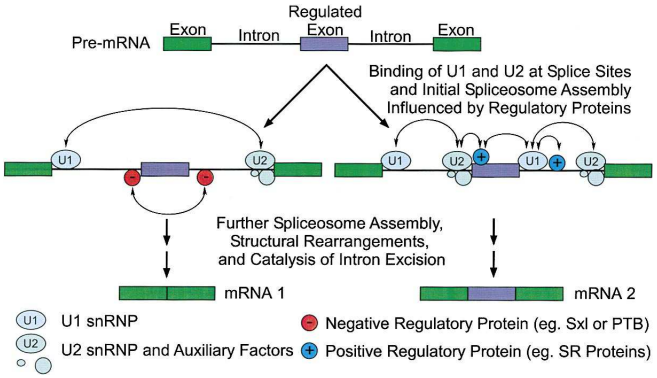

Figure 4.2: Alternative splicing (figure reproduced from [11]). Top: pre-mRNA molecule consisting of two unregulated introns, one regulated and two unregulated exons. The regulated exon may be spliced out depending on environmental conditions or cell type. Middle: regulatory proteins determine where splicing of the pre-mRNA molecule will occur. Only parts of pre-mRNA residing between U1 and U2 snRNPs (seen in the downstream direction) are removed. Left: negative regulatory factors prevent binding of U1 and U2 upstream and downstream, respectively, of the regulated exon. The latter will be cleaved together with the introns. Right: positive regulatory proteins lead to additional binding of U2 upstream and U1 downstream of the regulated intron. As only parts of pre-mRNA residing between U1 and U2 snRNPs are removed, the regulated exon remains in the molecule. Bottom: the two possible types of mature mRNA (denoted as mRNA 1 and mRNA 2) obtained by alternative splicing.

genetic and proteomic research has revealed, however, that between 50 to 70 % [63] of all human genes have more than one spliced variant, i.e. the pre-mRNA transcribed from them can be spliced alternatively into several different types of mature mRNAs. Thereby, the number of possible mature mRNAs stemming from the same gene varies widely, often ranging from three to seven in mammals, but sometimes even hundreds [63] up to 38000 (DSCAM in Drosophilia [84]) spliced forms can result from one single gene.

The effect of altered mRNA splicing on the structure and function of the encoded protein is remarkable [11]. Often, proteins which are encoded by the same gene differ in the number of amino acids by some hundreds if their pre-mRNA is spliced differently. This structural diversity is also being reflected in the function of the related proteins. For instance, changes in splicing have been shown to determine the ligand binding of growth factor receptors and cell adhesion molecules, to alter the activation domains of transcription factors, or to determine the subcellular localisation of the encoded protein as well as its phosphorilation by kinases (see [11] and references therein). Hence, it seems that the comparatively low number of genes found in the human genome is compensated by the large variety of mature mRNAs constructed by alternative splicing.

The principles of alternative splicing are depicted in Fig. 4.2. At its top a pre-mRNA molecule containing one regulated and two unregulated exons is shown. Depending on the cell type and/or the environmental conditions the regulated exon is either treated as an intron and cleaved from the pre-mRNA (left branch of in Fig. 4.2) or it remains as part of the mature mRNA molecule (right branch of Fig. 4.2). I.e., in this example two different types

of mature mRNA can be alternatively spliced from the same gene (bottom of Fig. 4.2).

Currently, it is not fully understood how cells steer alternative splicing at the molecular level, though, some basic mechanisms have already been identified. Again, as in the control of transcription initiation special regulatory proteins seem to play an important role (see Fig. 4.2 middle) [11]. These proteins are capable of detecting and binding to intron-exon boundaries sequence specifically and may interact with snRNPs participating at spliceosome assembly. Thereby, two types of proteins are distinguished: positive regulatory proteins enhance the binding of the snRNPs U1 and U2 to sites that are otherwise poorly recognised, and thus initiate the splicing process at points of the pre-mRNA which would otherwise remain uncleaved. Negative regulatory proteins, in contrast, mask intron-exon boundaries or function by hampering the build-up of the splicesome after U1 or U2 have bound.

Often, however, these regulatory proteins seem to function only if further proteins are bound to the enhancer or repressor regions of the gene to be expressed [63]. Furthermore, the secondary structure of the pre-mRNA molecule as well as the transcription rate seem also to be important factors determining where a pre-mRNA molecule is spliced.

4.1.4 Translation

Even if the events during translation do not influence the data recorded in microarray experiments directly they will still be discussed succinctly for the

sake of completeness. Translation is the last event of gene expression during which proteins are synthesized according to the genetic information found in (mature) mRNA. In prokaryotes translation often starts already during the transcription process while in eukaryotes mRNA is only translated after being completely transcribed, spliced and removed from the nucleus.

Two major players, the ribosomes as well as tRNA, participate at the translation process. Ribosomes consist of two separated subunits, a small and a large one, when not involved in translation. Each of these subunit is made of rRNA combined with a set of ribosomal proteins. This concept holds for both prokaryotic and eukaryotic ribosomes, though, the latter are somewhat larger than the first.

Similarly, tRNA molecules are nearly identical in eukaryotic and prokaryotic cells. They are about 75 to 90 base pairs long and are somewhat unique in that they contain several unusual nucleotides, such as inosine, pseudouridine, and hypoxanthine. tRNAs have two important functional domains called anticodon and amino acid binding site, respectively. The anticodon consists of three unpaired nucleotides which are used to base pair with the complementary codon on mRNA during translation. The amino acid binding site is found at the 3' end of the tRNA molecule and is used to charge the tRNA molecule with specific amino acids. Thereby, a tRNA molecule with a particular anticodon sequence binds to exactly one type of amino acid thus maintaining specificity of the genetic code (e.g. the tRNA with AGU as an anticodon sequence will only be charged with the amino acid serine).

The first step of the translation process is the binding of the small ribosomal subunit to some (mature) mRNA molecule. In prokaryotes, this subunit

attaches itself to the so-called Shine-Delgarno sequence (AGGAGG), which is part of every prokaryotic mRNA. Eukaryotic mRNAs are lacking this sequence but they are recognised by their 5' cap (cf. 4.1.2.2) to which the small ribosomal subunit binds to. After this binding the small subunit scans along the mRNA in the 5' to 3' direction until it encounters the start codon AUG (cf. A.2.2).

Once arrived there, a special tRNA molecule having UAC (i.e. the complement of AUG) as its anticodon sequence joins in. In prokaryotes this tRNA molecule is charged with a special amino acid called formyol methionine while in eukaryotes methionine is found. This tRNA molecule attaches itself to the mRNA molecule by base pairing of its anticodon with the start codon. Upon this binding the large ribosomal subunit appears and attaches itself to the small subunit-mRNA-tRNA-complex.

Next, the codon downstream of the start codon is bound to by its complementary (charged) tRNA molecule such that two amino acids are now attached to the mRNA molecule via their tRNAs. These two amino acids are then linked together by peptide bonds with the aid of an enzyme called peptidyl transferase.

In the next step, the ribosome shifts over one codon (in the 5' to 3' direction) and the tRNA molecule complementary to the third codon binds to the mRNA molecule. Again, peptide bonds between the newly brought in amino acid and the already existing ones are established. This process repeats itself until one of the termination codons (cf. A.2.2) is reached. These termination codons are bound to by so-called release factors which cause the translation complex (i.e. the constructed protein) to separate from

the mRNA molecule and thus terminate the translation process. However, mRNA molecules are usually transcribed several times, i.e. after or often even during the translation by the first ribosome a further ribosome assembles at the mRNA molecule and synthesizes another protein. How often the translation process is repeated depends on the lifespan of the mRNA molecule which is encoded by the length of its polyadenine tail (cf. 4.1.2.2).

4.2 Summary

This chapter provided a detailed introduction into the principles of molecular genetics of both prokaryotes and eukaryotes. First, similarities and differences between prokaryotic and eukaryotic cells were reviewed for convenience. Afterwards, the chemical structure of DNA and RNA was elucidated and it was shown how genetic information is stored in the form of codons in these two amino acids. Starting from the Central Dogma of Molecular Biology a precise definition of the term "gene" was given and it was shown how these genes are expressed. Thereby, emphasis was put on the transcription process and on alternative splicing as well as on the regulatory mechanisms steering these processes. Finally, the translation process of (mature) mRNA into proteins was discussed briefly.

Chapter 5

Microarrays

Microarrays are the state of the art method used to measure gene expression at the molecular level. They owe much of their popularity to the fact that they allow the quantification of the expression of tens of thousands of genes in parallel. This feature is especially appealing for geneticists as it enables them to monitor genomewide expression patterns in a single experiment. In practise microarray experiments are often used to detect differences in gene expression between healthy and cancer cells of the same type, between identical cells which are exposed to varying stimuli as well as between different types of cells.

Basically, microarrays measure the expression of genes by determining the amounts of mRNA transcribed from them. Thereby, use is made of the Central Dogma of Molecular Biology stating that information stored in a gene can only be used by a cell if it is first transcribed to mRNA (or simply to RNA in prokaryotes). As already described in 4.1.4 the so obtained mRNA molecule is sent to one of the ribosomes where its information is used to

synthesize protein until it gets degraded. Obviously, the more of a certain type of mRNA molecules are present in a cell, the larger will be the rate at which its corresponding proteins are synthesized. Hence, an elegant way to assess the activity of a gene is to measure how much mRNA transcribed from it can be found in a cell.

The chapter at hand gives a general introduction into the experimental procedure used to measure mRNA levels by means of microarray chips. Furthermore, the fabrication of the two most common types of microarrays will be explained and it will be shown which techniques are applied to obtain arrays with high sensitivity, specificity and robustness. Finally, a the chapter ends with a general discussion of the limits of the microarray technology.

5.1 The microarray experiment

In Fig. 5 a flowchart of the typical steps carried out during a microarray experiment is shown. As the quantities of mRNA produced by a single cell are far too small to be analysed quantitatively a huge amount of cells has to be breeded in the first step of the microarray experiment. Thereby, care has to be taken that all cells are treated equally and are exposed to the same stimuli in order to ensure that all of them produce the same levels of the individual types of mRNA.

In the second step the mRNA molecules are extracted from their cells by means of various organic solvents. Being rather fragile and easily degraded by enzymes common in many biological solutions these mRNA molecules cannot be used directly in the further course of the experiment. Hence, stabler

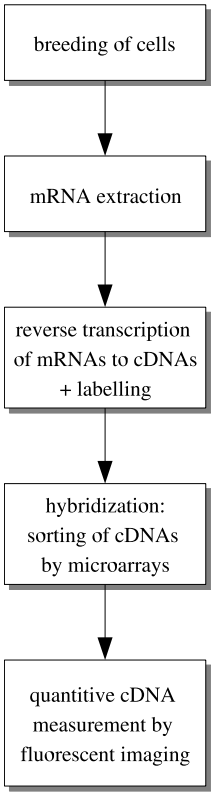

Figure 5.1: Flowchart of the typical steps carried out during a microarray experiment

copies of them have to be made in a process called reverse transcription [61] (step three in the flowchart of Fig. 5.1). Thereby, each mRNA molecule is transformed into a stable DNA molecule complementary to it. Note, that because of the specificity of base pairing no genetic information is lost in this process. During reverse transcription, the obtained complementary DNA (cDNA) molecules are also labelled with a fluorescent dye which is needed to detect them later on.

Usually, cells transcribe many genes in parallel leading to a large number of different types of cDNA which have to be sorted before their individual quantities can be determined. This sorting is achieved by means of a mi-

croarray chip in the fourth step, called hybridisation step, of the experiment. These chips have tens of thousands of spots on their surface which are arranged in a well-defined grid like structure. At each of these spots a large number of single stranded DNA (ssDNA) molecules is found. Thereby, all ssDNA molecules located at the same spot are identical while the ssDNAs at different spots vary in their sequence. The sequences of the ssDNA molecules at the individual spots are not arbitrary but are chosen such that they equal the sequences of known or putative mRNA molecules of the organism under investigation. With the number of spots being very large, this means that the majority of all potentially transcribed mRNA molecules of an organism are represented on the microarray chip. Hence, for (almost) each type of cDNA molecules obtained by reverse transcription complementary ssDNA molecules are found at one of the spots of the microarray.

Over the surface of the microarray chip a solution of the cDNA molecules obtained by reverse transcription is poured and pumped around. If a cDNA molecule comes across a dot containing ssDNA molecules complementary to it it hybridises to one of them. Thus, after some time pumping, (almost) all cDNA molecules will have bound to their complementary ssDNA molecules on the individual spots of the microarray. Obviously, the more of a certain type of cDNA was present in the solution the more occupied the dot containing its complementary ssDNA will be.

This occupancy is detected by fluorescent imaging in the last step of the microarray experiment. Thereby, typically a laser with a frequency in the ultraviolet region (e.g. an argon laser) is used to excite the fluorescent dyes which were attached to the cDNA molecules after reverse transcription. This

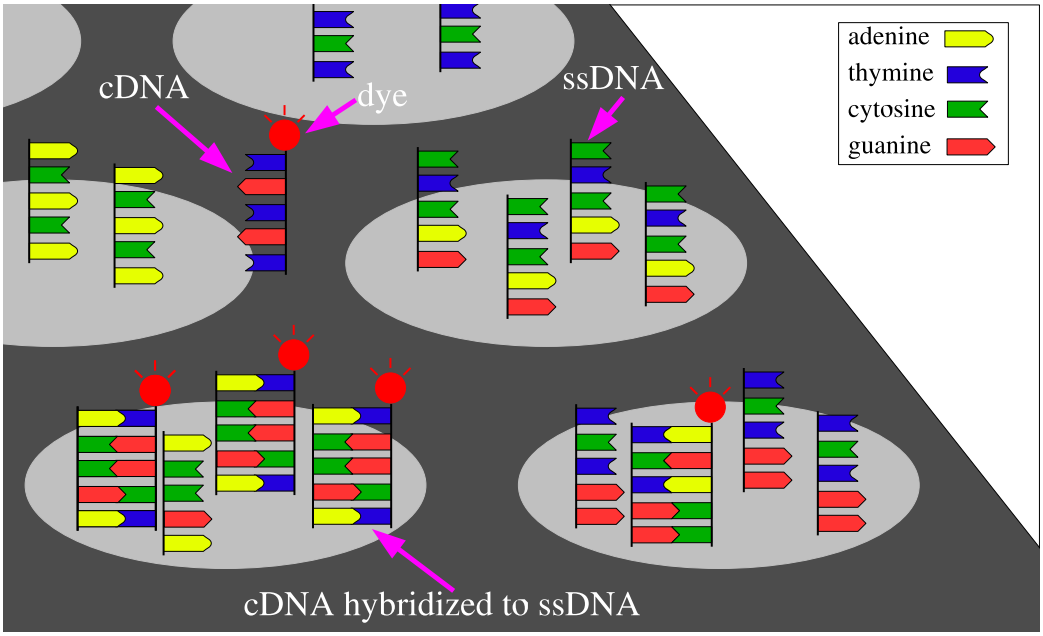

Figure 5.2: Zoom in onto a microarray chip after application of cDNA solution. Grey circles: spots on surface of the chip used for detecting individual types of cDNA. Note: each spot contains only ssDNA of the same sequence. Also shown: single cDNA (with sequence TGTGT) labelled with dye which is still floating over microarray surface as well as cDNA hybridised to ssDNA.

laser is sequentially targeted at all the individual spots of the microarray chip while a detector measures and records the photons emitted by the excited dyes. Thereby, the number of photons detected depends linearly on the number of cDNA molecules bound to the spot excited by the laser. As the genetic sequence of the ssDNAs of each spot are known this measurement eventually reveals how much cDNA of a specific type was present in the solution. Thus, also the number of mRNA molecules produced by the cells under investigation is known.

5.2 Reverse Transcription

As already mentioned in the introduction of this chatper mRNA molecules are rather fragile and easily destroyed by enzymes common in many biological solutions. Hence, more stable copies of them have to be made by reverse transcription [61]. Thereby, the original mRNA molecules are used as templates to synthesize single stranded DNA molecules complementary to them. Because of the specificity of base pairing these complementary DNA (cDNA) molecules contain the same genetic information as the original mRNA molecules. Furthermore, only one cDNA molecule is transcribed from each mRNA molecule such that the information about the amounts of each type of mRNA is also preserved. The so-obtained cDNA molecules are much stabler than mRNA and are hence preferred in many biological analysis methods.

Technically, reverse transcription typically consists of the following steps:

1. Denaturation of mRNA. Before the copying process can be initiated the individual mRNA molecules need to lose their secondary or tertiary structure (i.e. their local or global spacial structure). This is achieved by heating the mRNA up to high temperatures for a short period of time (typical values: 95 ℃ for 5 minutes).

2. Snap chill to low temperatures. In the this step the tubes containing mRNA are put on ice and thus cooled down to about 4 ℃. This step has to be carried out quickly as otherwise the mRNA molecules in the tube could start to renaturate, i.e. to regain their spacial structure.

3. Primer hybridisation and elongation. Similar to gene transcription an enzyme called reverse transcriptase (e.g. avian myeloblastosis virus, moloney murine leukemia virus or thermus thermophilius) is needed to catalyse the synthesis of a cDNA strand from an original mRNA molecule. However, reverse transcriptase cannot start the synthesis from scratch i.e. some bases of the mRNA molecule to be copied must already be bound to by their complementary counterparts. For this purpose so-called primers are used. In the case of eukaryotic cells these primers often consist of a short single strand of thymine which hybridises to the polyadenine tail of the mRNA molecule. To these thymine bases reverse transcriptase adds nucleotides complementary to those found in the original mRNA molecule in the 3' to 5' direction (of the mRNA molecule) and connects them by phosphodiester bonds. Thereby, also fluorescent dyes can be incorporated. Afterwards, reverse transcriptase cleaves the bondings between cDNA and mRNA and starts to disintegrate the mRNA molecule. Thus, a complementary DNA molecule is synthesized while the original mRNA molecule vanishes.

Finally, it should be noted that the temperatures at which primer hybridisation and elongation occur vary significantly. While elongation is typically most efficient between 37 ℃ to 42 ℃ primer hybridisation should usually occur at about 70 ℃. However, the latter value varies widely depending on the type of primer used.

5.3 Fabrication of microarry chips

Basically, microarrays can be fabricated by two different methods which distinguish themselves by how the ssDNA molecules are applied to the surface of the microarray chip. In the case of so-called spotted microarrays, prefabricated ssDNA molecules are deposited by a robot on the surface of the substrate whereas in the case of *in-situ* synthesis the ssDNAs are synthesized photochemically on the chip. Both methods will be the subjects of the following sections.

5.3.1 Spotted microarrays

Before a spotted microarray chip can be fabricated, large amounts of the ssDNAs which will later be deposited on the surface of the substrate are needed. These large amounts of ssDNA are synthesized in a process called polymerase chain reaction (PCR), a very efficient method capable of creating billions of copies of specific fragments of DNA from a single DNA molecule (see Appendix B for details).

Even if multiplication by PCR is a very efficient method it still has to be repeated for each of the many thousands of different types of ssDNAs required on a microarray chip for genomewide gene expression analysis. Accordingly, the production of the various ssDNAs is the major cost factor in spotted microarray fabrication.

Once sufficient amounts of each type of ssDNA have been synthesized they have to be deposited in a grid-like manner onto the surface of the microarray chip whereas the position of each type of ssDNA is recorded in a data file. The

deposition of the individual ssDNAs onto the chip is carried out by a robot which, in the simplest case, dips a thin pin or needle into a solution containing a specific type of ssDNA molecules. This pin is then brought to a well defined position on the grid where it gently touches the microarray's surface thus releasing a small bead of ssDNA solution. These two steps are repeated until all types of ssDNA molecules are placed on their individual position on the grid [31]. Apart from thin pins and needles also instruments similar to fountain pens are used in contact spotting techniques. These pens are filled successively with the individual ssDNA solutions which are afterwards set under constant pressure. Once such a pen is brought into contact with the surface of the microarray chip it releases a small amount of the type of ssDNA solution currently stored in it [61].

Apart from contact also noncontact spotting techniques have been developed in recent years making use of the piezoelectric capillary effect or of technologies borrowed from ink-jet printing. In both cases small amounts of each type of ssDNA solution are shot on the surface of the microarray chip [61] thus producing the desired grid-like distribution of the individual ssDNAs.

In order to detect the expression of a large number of genes in a single experiment as many spots as possible are tried to be applied onto the surface of the microarray chip. This can be facilitated by coating the surface with silicon hydrides as they repel water and thus prevent intermixing of neighboring spots on the grid. With such coatings spots with a diameter as small as 2 μm can be achieved [61] and up to 30000 of such spots are found on modern spotted microarry chips [24].

After the ssDNA has been deposited it must be fixed onto the surface in order to maintain the integrity of each spot and also to prevent the ssDNA from washing off during the various steps carried out in processing the array later on. This immobilisation of ssDNA is usually achieved by air drying which is then commonly followed by ultraviolet irradiation for DNA fixation. After these steps the spotted microarray chip is ready for use.

5.3.2 *In-situ* synthesis

During array fabrication based on *in-situ* synthesis the ssDNA strands are not produced separately but are synthesized directly onto the surface of the microarray chip [6]. The synthesis is achieved by means of photolitography, a well known technique in the field semiconductors which allows the synthesis of ssDNA strands by adding, step by step, single nucleotides together. Hence, no cloning or PCR of DNA fragments are necessary in *in-situ* synthesis.

The basis upon which the ssDNA molecules will be synthesized is usually a glass matrix called wafer. Before any nucleotides can be applied to the waver it has to be activated first in a process called silanation. Thereby, the waver is bathed in silane which attaches itself onto the glass surface of the waver by its Si atom (see Fig. 5.3 a). This step is necessary as many molecules cannot bind to glass by themselves. To each of these Si molecules a so-called linker combined with a photosensitive molecule is attached. The linkers are needed later on to allow the first nucleotides to bind to the surface of the microarray whereas the photosensitive molecules are used to regulate the build up of the individual ssDNAs in the further course of the synthesis.

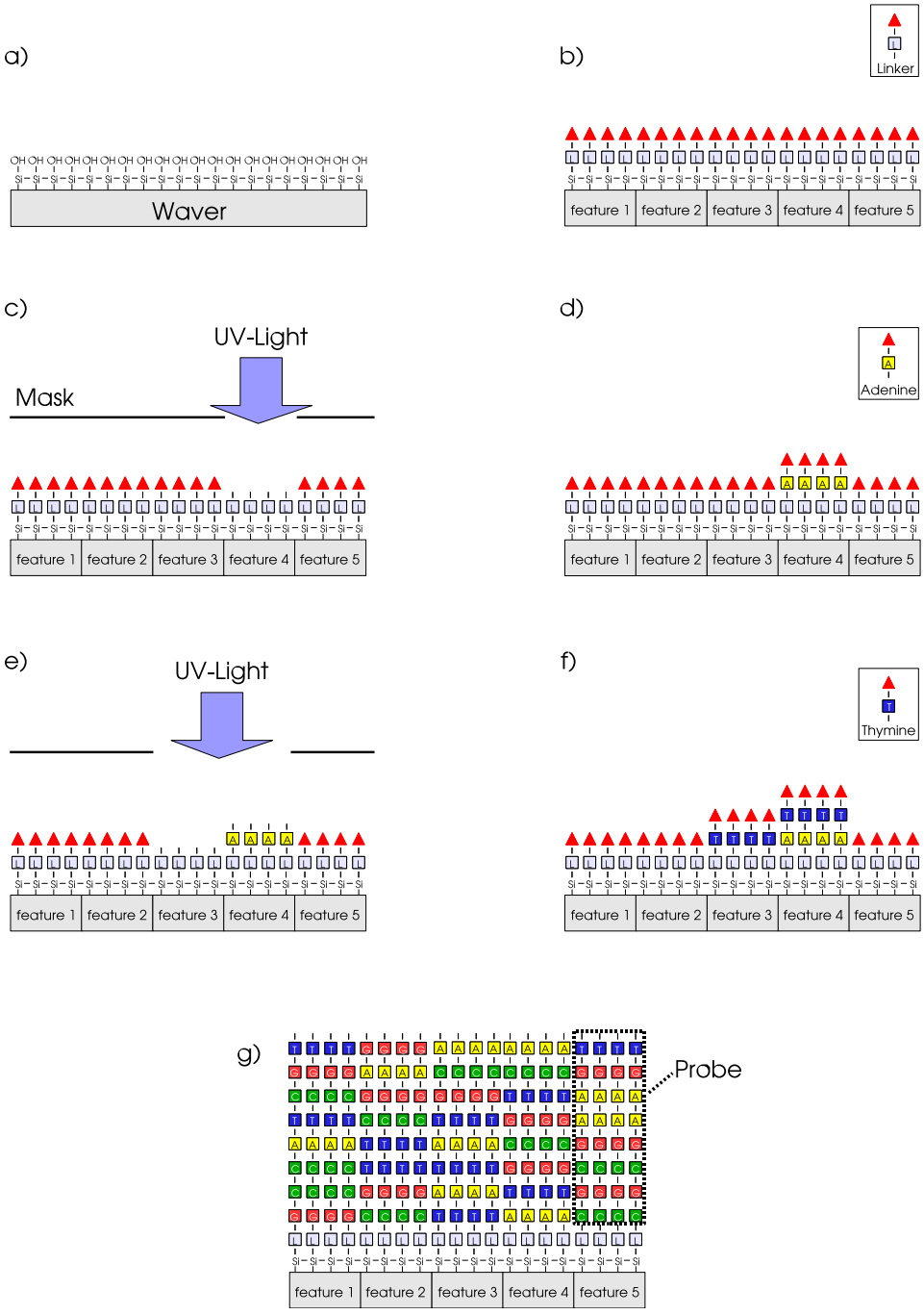

Figure 5.3: *In-situ* synthesis of ssDNA by means of photolitography. a) Waver after silanation. b) Linker molecules (depicted as a rectangles containing an L) combined with photosensitive molecule (red triangle) are added to the silane molecules. c) Photolitography: UV light destroys the photosensitive molecules at feature 4 as it is not covered by the mask. d) The new nucleotides (here adenine) together with their photosensitive cap are added to feature 4. e) A new mask is used to remove (by UV light) photosensitive molecules of features which are to accept the next type of nucleotides. f) The second type of nucleotides (here thymine) is added. g) Repetition of steps e) and f) leads to the build up of the individual ssDNA strands at each feature. ssDNAs with the same sequence are collectively referred to as a probe.

At this stage of the process the waver is equally subdivided into so-called features, i.e. into equally sized regions where identical ssDNA molecules are to be synthesized (Fig. 5.3 b). Although the size of such a feature is usually only $5 \times 5\ \mu m^2$ it will still contain millions of ssDNA strands at the end of the *in-situ* synthesis. In other words, these features will be the counterparts of the spots found on the surface of spotted microarrays.

In the next step the microarray is prepared to accept the first set of nucleotides. For this purpose a mask is centred over the chip which contains little "windows" which are designed to only let light through to the specific features chosen to receive the next nucleotide. This mask is illuminated by ultraviolet light which destroys the photosensitive molecules at uncovered features on the chip (Fig. 5.3 c). Hereupon the microarray is washed with a solution containing only one type of nucleotides which are capped with photosensitive molecules themselves (see insert in Fig. 5.3 d). These nucleotides can only bind to linkers which are not protected by photosensitive molecules, i.e. they only add to features which were exposed to ultraviolet light before (Fig. 5.3 d). Thus, it can be regulated precisely which nucleotides bind to which feature. Furthermore, as the newly added nucleotides are capped by photosensitive molecules all features of the chip will be capped after nucleotide binding as well. The next type of nucleotide is then added in the same way: first, a new mask is needed in order to irradiate only those features with ultraviolet light which are to obtain new nucleotides (Fig. 5.3 e); second, the microarray is washed in a solution containing the new type of nucleotides to be added (Fig. 5.3 f).

These steps are repeated until at each feature the ssDNAs with the desired

sequence are synthesized. Finally, the photosensible molecules are removed from the endmost layer of nucleotides in the last step of the synthesis process. Note, that collectively, all identical ssDNAs at a feature are often referred to as probe in the context of microarrays fabricated by *in-situ* synthesis (Fig. 5.3 g).

5.4 Differences in spotted and *in-situ* synthesized microarray experiments

In order to acquire high quality data in a microarray experiment the used procedure must guarantee

1. a high sensitivity to detect poorly expressed transcripts in a complex background,
2. a high specificity to distinguish between transcripts which differ only in a small number of nucleotides,
3. a high reproducibility of the data.

As will be described in the following different approaches and experimental procedures are used in spotted and *in-situ* microarray experiments in order to achieve these three goals.

5.4.1 Particularities of spotted microarrays

One of the major causes for limited sensitivity in experiments with spotted microarrays is background fluorescence stemming from either the support or

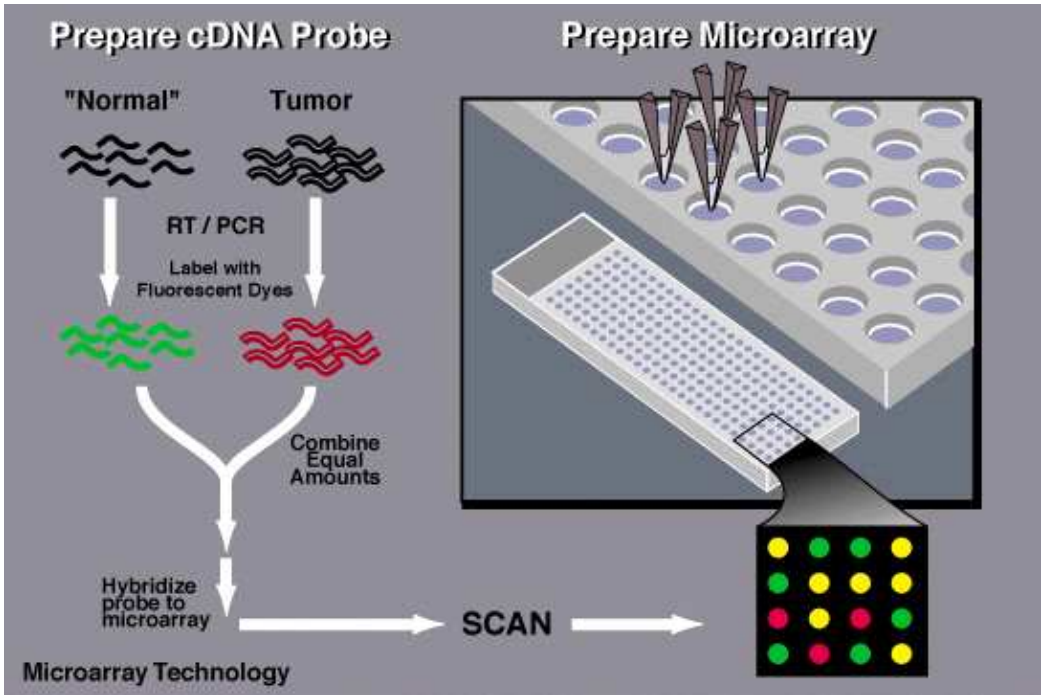

Figure 5.4: Flowchart of a comparative spotted microarray experiment (figure reproduced from [45])

from any cDNA molecules which attached themselves unspecifically to the surface of the chip. While unspecific cDNA binding can hardly be suppressed fluorescence originating from the support can be reduced drastically by coating it with silicon hydrades [61]. As already mentioned in 5.3.1 this coating also repels water thus helping to obtain very small spots on the microarray. Accordingly, such coatings are now found at almost all spotted microarray chips.

A main reason for lowered specificity is cross-hybridisation, i.e. the hybridisation of cDNA to noncomplementary ssDNA. This phenomenon often occurs between ssDNA and cDNA which are complementary expert for only a small number of bases. Thereby, the length of the ssDNAs plays an im-

portant role: the longer the ssDNA molecules are, the better hybridisation deficiencies caused by a small number of noncomplementary bases can be compensated. Hence, the length and sequence of the ssDNA molecules applied to the chip is chosen deliberately in order to avoid cross-hybridisation.

Another difficulty arising in experiments with spotted microarrays is the determination of absolute cDNA levels at the individual spots. The problem appears as different types of cDNA vary in the experimental conditions needed for optimal hybridisation. Hence, not all sorts of cDNA will bind efficiently to their corresponding spots such that a too low level of them is detected during fluorescence imaging. This problem also affects the reproducibility of the experiment as slight fluctuations in experimental conditions may lead to varying hybridisation efficiencies of the individual cDNAs and thus to different results.

In order to circumvent this issue researchers use cohybridisation to measure cDNA stemming from the cell type under investigation relative to cDNA levels originating from a reference cell type. These reference cells are usually healthy cells of the same type as the cells under investigation and are breeded under some standardised conditions.

The basic steps in these experiments are the following (Fig. 5.4): first, the mRNAs are extracted from both cell types and are kept separated. Then, during reverse transcription the cDNA molecules stemming from the first cell type are labelled with a different dye than the cDNA molecules of the second cell type. Thereby, usually cyanine 3 (green) is used to label the reference cells, while e.g. tumorous or stimulated cells are labelled by cyanine 5 (red). The differently labelled cDNAs are mixed and applied jointly to

the microarray chip where they hybridise with the ssDNAs on the individual spots. The particular levels of cDNA at each spot are measured by means of two lasers which emit light at frequencies suitable to excite either cyanine 3 or cyanine 5. Each of these lasers is used separately to record a fluorescence image of the microarray, i.e. altogether two fluorescence images are obtained whereas one of them consists of green spots stemming from cyanine 3 fluorescence while in the other red spots originating from cyanine 5 fluorescence are found. Finally, a computer is used to merge these two images into a new image whereas each spot in the new image is obtained by additively mixing the colours of the corresponding spots in the original images. This leads to the typical fluorescence pattern as shown in Fig. 5.5 where four types of spots can be distinguished:

1. At black spots no hybridisation occurred, i.e. the related gene was not expressed by neither of the two cell types.

2. Green spots indicate that the corresponding gene was only expressed by healthy or standard cells, respectively.

3. Red spots indicate that the corresponding gene was only expressed by tumorous or stimulated cells, respectively.

4. Yellow spots arise when both types of cells expressed a certain gene. The different shades of yellow correspond to the ratio between cDNA stemming from the cells under investigation and the reference cells.

Hence, gene expression is not determined absolutely but regarded with respect to some reference cells. Regarding ratios instead of absolute values also

leads to much robuster results as both the cDNA from the cells under investigation as well as from the reference cells are equally affected by changing experimental conditions. Thus, their ratios remain constant even if the experimental is repeated under different conditions. This, of course, only holds for spots where cDNAs of both types of cells have bound to (i.e. only for the yellow spots in the above listing).

Furthermore, using cohybridisation also reduces saturation issues which appear when less ssDNA molecules are present on the chip than corresponding cDNAs. To illustrate this imagine that the number of ssDNAs on a certain spot on the chip is smaller than the number of complementary cDNA molecules which have to be detected. Furthermore assume that the cDNAs were produced by both types of cells which means that a certain fraction of cDNA molecules will be labelled with a green dye while the remaining ones are labelled with a red one. As both red and green labelled cDNAs bind equally well to their complementary ssDNAs the ratio between the red and the green cDNAs bound to the spot will equal the ratio of overall red and green cDNAs as found in the solution. Thus, even if not all cDNA molecules can bind to their corresponding spot their ratio can still be determined by means of this method. However, absolute transcription values cannot be obtained by this procedure.

5.4.2 Particularities of *in-situ* synthesized microarrays

In the sequel, techniques developed by Affymetrix® [4] will be presented which help to increase the sensitivity and specificity in experiments car-

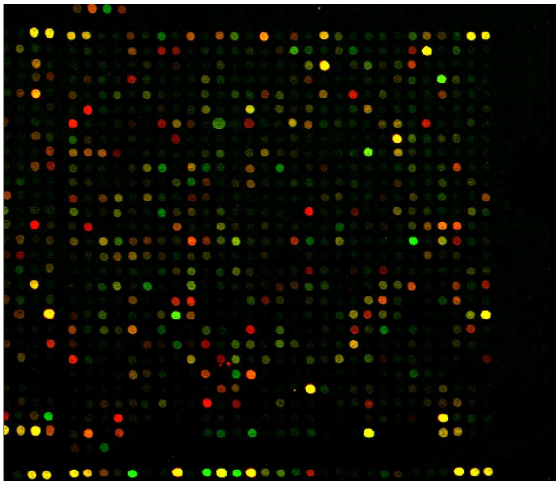

Figure 5.5: Fluorescence image obtained by comparative microarray experiment (figure reproduced from [34])

ried out with *in-situ* synthesized microarray chips. As will be shown, these techniques also allow the robust determination of absolute transcription levels such that results stemming from different experiments can be compared quantitatively with each other.

While in spotted microarrays often ssDNAs with a length of up to 1500 base pairs are found only 25 base pairs long strands (called 25-mers in the sequel) are used in Affymetrix® microarrays. At each probe, these 25-mers are chosen such that they are complementary to a sequence which can only be found in one of the various types of cDNA to be detected. Such short ssDNAs have the advantage that a single mismatch between them and an otherwise complementary cDNA segment is already sufficient to destabilise hybridisation. Thus, cross-hybridisation effects can be reduced by these 25-mers and a high level of specificity is achieved.

Furthermore, 22 different probes, which are collectively referred to as a probe set, are used to detect a certain type of cDNA molecules. These probes

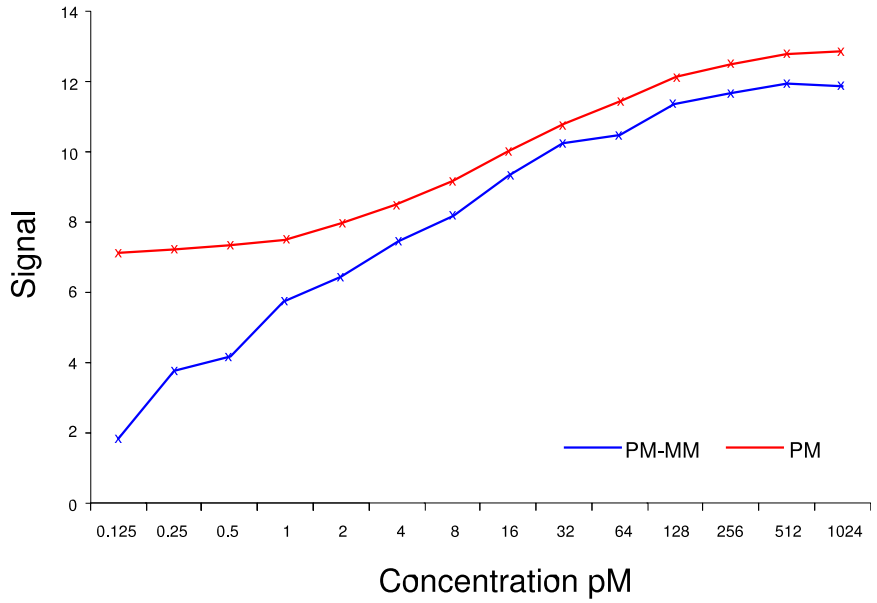

Figure 5.6: Comparison of sensitivity using PM probes only or PM-MM probe pairs [4].

differ in their particular sequences, i.e. each of them binds to another specific part of the cDNA molecule to be detected. Using data from several of such probes per cDNA increases the specificity and sensitivity and leads to robuster measurements as outliers at single probes can be detected. Furthermore, using several probes to measure a single type of cDNA also averages out differences in hybridisation efficiencies.

As in the case of spotted microarrays the sensitivity of *in-situ* synthesized microarrays is mostly limited by background signals. To cope with this problem Affymetrix® always synthesizes pairs of probes onto the microarray which consist of the normal, perfect match (PM) probe used to detect a specific type of cDNA and a mismatch (MM) probe.

The 25-mers of this MM probe differ from the 25-mers of the PM probe only in a single, homomeric (nucleotide mismatch that contains the comple-

mentary base to the original) base at the 13th position while all other bases are identical. Hence, it can be assumed that the MM probe hybridises to nonspecific sequences about as effectively as its PM counterpart, allowing spurious signals, e.g. from cross-hybridisation or stray light, to be efficiently quantified and to be subtracted from the fluorescence levels measured at the PM probes. As can be seen in Fig. 5.6 this PM-MM strategy is especially advantageous if very low levels of cDNA are to be detected, i.e. when the singal to noise ration at the PM probes is already very low. Thus, transcripts representing 1:100000 - 1:300000 of total transcripts can be detected with modern chips.

By means of the PM-MM strategy also detection calls can be assigned to each gene. These calls classify a gene as either present (P), which means that its expression level is well above the minimum detectable level, absent (A), which means that its expression level is below this minimum, or marginal (M), which means that the expression level of the gene is close to the minimum detectable level.

In the following the algorithms used by Affymetrix® to determine signal levels as well as detection calls will be presented. Furthemore, the normalisation and scaling techniques needed to compare data from different microarray experiments will be discussed.

5.4.2.1 Signal Calculation

The singal value to be calculated should represent the actual amount of cDNA bound to a spot as accurately as possible. For this purpose, any spurious signals originating from cross-hybridisation or stray light have to be

quantified and multiple probes per cDNA type have to be used such that a certain robustness of the obtained data is assured. As mentioned above, MM probes are used to measure background signals which lead to elevated fluorescence levels at the corresponding PM probes. MM values smaller than the corresponding PM values are considered to be good estimates of these background signals and can be used directly in the further computations. However, sometimes MM values larger than their corresponding PM values are measured which would mean that the detected spurious signal is larger than the spurious signal plus the signal stemming from the bound cDNA. This discrepancy often appears when the solution applied onto the chip contains a hitherto unknown type of cDNA which is complementary to one of the MM probes or if mutations lead to modifications in the cDNA transcript to be detected such that it rather binds to the MM than to the PM probe [3]. In such cases, the MM probe functions as a PM probe, i.e. it cannot be used to quantify any background signals. Hence, the obtained MM value has to be replaced by a so-called idealised mismatch (IM) value, an estimate which is based on the knowledge of the whole probe set or on the behaviour of probes in general (see Fig. 5.7). This IM is determined according to the three following rules:

1. MM values smaller than their corresponding PM values are considered to be good estimates of spurious signals. Hence, the IM value is simply set equal to the MM value in such cases:

$$IM_{i,j} = MM_{i,j} \text{ for } MM_{i,j} < PM_{i,j}. \tag{5.1}$$

Here and in the following, the index pair (i, j) denotes the j-th probe in the i-th probe set whereas $1 \leq i \leq m$ and $1 \leq j \leq n$ if m probe sets consisting of n probes pairs where synthesized onto the chip.

2. If only a small number of MM values in a probeset is larger than their corresponding PM values a specific background (SB) can be estimated for the probe set as follows:

$$SB_i = T_{bi}(\log_2(PM_{i,j}) - \log_2(MM_{i,j})) \text{ for } j = 1, \ldots, n, \tag{5.2}$$

whereas T_{bi} is a one step Tukey's biweight estimate similar to the weighted mean. Figuratively, this SB_i is a typical log ratio of PM to MM values in a given probe set. If the values of the probe set are generally reliable (up to a small number of exceptions where $MM \geq PM$) a large SB_i will be obtained which is then used to compute the IM:

$$IM_{i,j} = \frac{PM_{i,j}}{2^{SB_i}} \text{ for } MM_{i,j} \geq PM_{i,j} \text{ and } SB_i > \tau_{contrast}, \tag{5.3}$$

whereas $\tau_{contrast}$ is a threshold representing the minimum value the SB_i must have in order to consider the values obtained from a probe set holistically as reliable ($\tau_{contrast} = 0.03$ default value used by Affymetrix®).

3. If the value obtained for the SB_i is very low (i.e. lower than $\tau_{contrast}$) the majority of the probes in the probe set are meaningless. In such

cases, more of the PM signal is used in the estimation of the IM value:

$$IM_{i,j} = \frac{PM_{i,j}}{2^{1+\frac{\tau_{contrast}-SB_i}{\tau_{scale}}}} \text{ for } MM_{i,j} \geq PM_{i,j} \text{ and } SB_i \leq \tau_{contrast}, \quad (5.4)$$

whereas τ_{scale} is the cutoff that describes the variability of the probe pairs in the probe set ($\tau_{scale} = 10$ default value used by Affymetrix®). Thereby, the least informative IM estimate is obtained which is only based weakly on the probe-specific data. Hence, genes measured by such probes are usually labelled with the detection call "absent".

Once the IM value has been determined for each probe pair on the microarray, the so-called probe value (PV) is computed. For this purpose, the estimate of the background, i.e. the IM value, is subtracted from the corresponding PM values

$$V_{i,j} = \max(PM_{i,j} - IM_{i,j}, \delta), \text{ default } \delta = 2^{-20}, \quad (5.5)$$

whereas the small parameter δ is needed for numerical stability in the further course of the analysis. From these $V_{i,j}$'s, the PV of each probe pair is obtained by taking the logarithm:

$$PV_{i,j} = \log_2(V_{i,j}), \ i = 1, \ldots, m, \ j = 1, \ldots, n. \quad (5.6)$$

Finally, the PVs of each probe set are averaged such that the signal log value (SLV) is obtained:

$$SLV_i = T_{bi}(PV_{i,1}, \ldots, PV_{i,n}). \quad (5.7)$$

The anti-log of this SLV_i (i.e. 2^{SLV_i}) is the desired robust and denoised estimate of the amount of cDNA bound to the i-th probe set.

5.4.2.2 Comparison Analysis

In comparison analysis the data obtained from two different microarray experiments, usually referred to as experiment and baseline, are compared quantitatively. Before the two data sets can be compared they have to be normalised or scaled first in order to correct any general variations between them. These general variations are mostly caused by biological or technical differences. Biological variations may originate from many sources like, e.g., fluctuations in the genetic background or in growth conditions, varying dissection techniques, or differences in time, weight, sex, age, and replication. On the other hand most of the technical variations may stem from fluctuating experimental variables like quality and quantity of hybridised cDNA, reagents, stain, or handling errors [5].

One way to cope with such variations between two microarray experiments is scaling, a method by which the averages of the experimental and the baseline dataset are brought to a common, user defined level Sc. For this purpose, a scaling factor $sf^{\{experiment\}}$ or $sf^{\{baseline\}}$, respectively, is determined for each dataset:

$$sf^{\{k\}} = \frac{Sc}{TrimMean(2^{SLV_i^{\{k\}}}, 0.02, 0.98)}, \; k \in \{baseline, experiment\}. \tag{5.8}$$

The used trimmed mean function ($TrimMean$) first removes the 2% largest and 2% smallest $SLV_i^{\{k\}}$'s in each dataset before calculating the mean of the

remaining $SLV_i^{\{k\}}$s. This procedure leads to a reduced sensitivity to outliers in comparison to the standard mean computation of the SLV_i's.

The scaling factors are used to scale the $SLV_i^{\{k\}}$'s in the corresponding data sets such that the so-called reported values $RV_i^{\{k\}}$ are obtained:

$$RV_i^{\{k\}} = sf^{\{k\}} \cdot 2^{SLV_i^{\{k\}}},\ k \in \{baseline, experiment\} \tag{5.9}$$

The so-obtained RV_i's have the desired trimmed mean value Sc, i.e.

$$TrimMean(RV_i^{\{k\}}, 0.02, 0.98) = Sc,\ k \in \{baseline, experiment\}. \tag{5.10}$$

If normalisation is used to adjust two data sets, only the experimental array is scaled by a factor nf such that its average intensity equals that of the base array:

$$RV_i^{\{experiment\}} = nf \cdot 2^{SLV_i^{\{experiment\}}} \tag{5.11}$$

$$RV_i^{\{baseline\}} = 2^{SLV_i^{\{baseline\}}} \tag{5.12}$$

whereas

$$nf = \frac{TrimMean(SLV_i^{\{baseline\}}, 0.2, 0.98)}{TrimMean(SLV_i^{\{experiment\}}, 0.2, 0.98)} \tag{5.13}$$

such that

$$TrimMean(RV_i^{\{experiment\}}, 0.2, 0.98) = TrimMean(RV_i^{\{baseline\}}, 0.2, 0.98). \tag{5.14}$$

As in the case of scaling the obtained values are again called reported values

(RV).

Once the scaling or normalisation factor is determined the individual probe pair values on each chip also have to be modified. These scaled probe values (SPV) are determined as

$$SPV_{i,j}^{\{k\}} = PV_{i,j}^{\{k\}} + \log_2(sf^{\{k\}}),\ k \in \{baseline, experiment\} \tag{5.15}$$

if scaling was used or as

$$SPV_{i,j}^{\{experiment\}} = PV_{i,j}^{\{experiment\}} + \log_2(nf) \tag{5.16}$$

$$SPV_{i,j}^{\{baseline\}} = PV_{i,j}^{\{baseline\}} \tag{5.17}$$

if normalisation was applied.

Based on these SPV's the data sets of the two microarray experiments can be compared at the individual probe level. For this purpose a probe log ratio (PLR) for each probe j in the probe set i on both baseline and experiment array is computed:

$$PLR_{i,j} = SPV_{i,j}^{\{experiment\}} - SPV_{i,j}^{\{baseline\}} \tag{5.18}$$

Next, the PLR values belonging to the same probe set are averaged such that the signal log ratio (SLR) is obtained for each probe set:

$$SLR_i = T_{bi}(PLR_{i,1}, \ldots, PLR_{i,n}). \tag{5.19}$$

These signal log ratios eventually indicate in which of the two experiments a

specific gene was expressed more. However, often geneticists do not compare expression levels by there SLV's but by fold changes (FC), which are derived from the SLV's as follows:

$$FC_i = \begin{cases} 2^{SLR_i}, & SLR_i \geq 0 \\ -2^{-SLR_i}, & SLR_i < 0 \end{cases} \tag{5.20}$$

Hence, if a positive fold change is computed for a certain probe set i, the corresponding gene was more expressed in the experiment than in the baseline dataset and vice versa.

5.4.2.3 Detection Calls

Another important advantage of using several probes to detect a single type of cDNA is that detection calls can be determined which tell the user if the intensity measured at a certain spot stems from a transcript or from background noise [7]. For the determination of these detection calls each probe pair in a probe set is evaluated by means of a discriminant score R_i which is defined as

$$R_i = \frac{PM_i - MM_i}{PM_i + MM_i}, \tag{5.21}$$

where PM_i and MM_i are the extracted intensities for the i-th perfect match probe and mismatch probe, respectively. A positive R_i value implies that the perfect match intensity PM_i is larger than MM_i, and the strength of detection ability of the i-th probe pair increases with R_i. A negative value implies that the mismatch intensity MM_i exceeds PM_i and reveals a poor detection ability of the i-th probe pair.

A detection p-value is determined next by means of the one-sided Wilcoxon signed rank test (see e.g. [65] for a detailed description) whereas the scores R_i computed for all probe pairs in a probe set are compared with a small user defined threshold τ, the null hypothesis being that no difference between the median discrimination score and τ exists. Figuratively, this means that each discriminant score which is larger than τ is seen as a "vote" for a significant signal while scores smaller than τ indicate that the signal is random. The p-value indicates which of the two cases were predominant in a given probe set.

Based on this p-value the detection calls are determined by means of two further, user defined thresholds α_1 and α_2 as follows:

Present	Marginal	Absent
$p < \alpha_1$	$\alpha_1 \leq p < \alpha_2$	$p \geq \alpha_2$

The impacts of altering α_1 or α_2 are as follows: by reducing α_1 the number of false detected calls but also the number of true detected calls is reduced. On the other hand increasing α_2 can reduce the number of false undetected calls as well as the number of true undetected calls.

5.5 Summary

In this section the basic concepts of detecting mRNA by means of microarray chips was presented. Furthermore, the fabrication procedures of two of the most commonly used microarrays, the spotted and the *in-situ* synthesized arrays were explained. As was shown, spotted microarrays are an efficient method when the gene expression of two types of cells are to be compared di-

rectly while problems arise when the total amount of cDNA bound to a spot is to be determined. In such cases, *in-situ* synthesized chips, with several perfect match - mismatch probe pairs per cDNA to be detected, are more appropriate. As was discussed, this multiprobe PM-MM strategy allows a robust detection of absolute cDNA levels but may also be used, after some normalisation steps, to compare several microarray experiments with each other. Even if the PM-MM strategy is a reasonable approach to deal with stray signals quantitatively it must be noted that noise is still a major problem in all microarray experiments. The reasons for noise a manifold, ranging from irregularities during cell breeding, mRNA extraction or reverse transcription to poorly chosen hybridisation parameters, fluctuating dye quality or impurities on the microarray chip [31]. Despite this problem, microarrays have become indispensable in many research laboratories and hospitals as they provide an overview of the expression of thousands of genes in a single experiment.

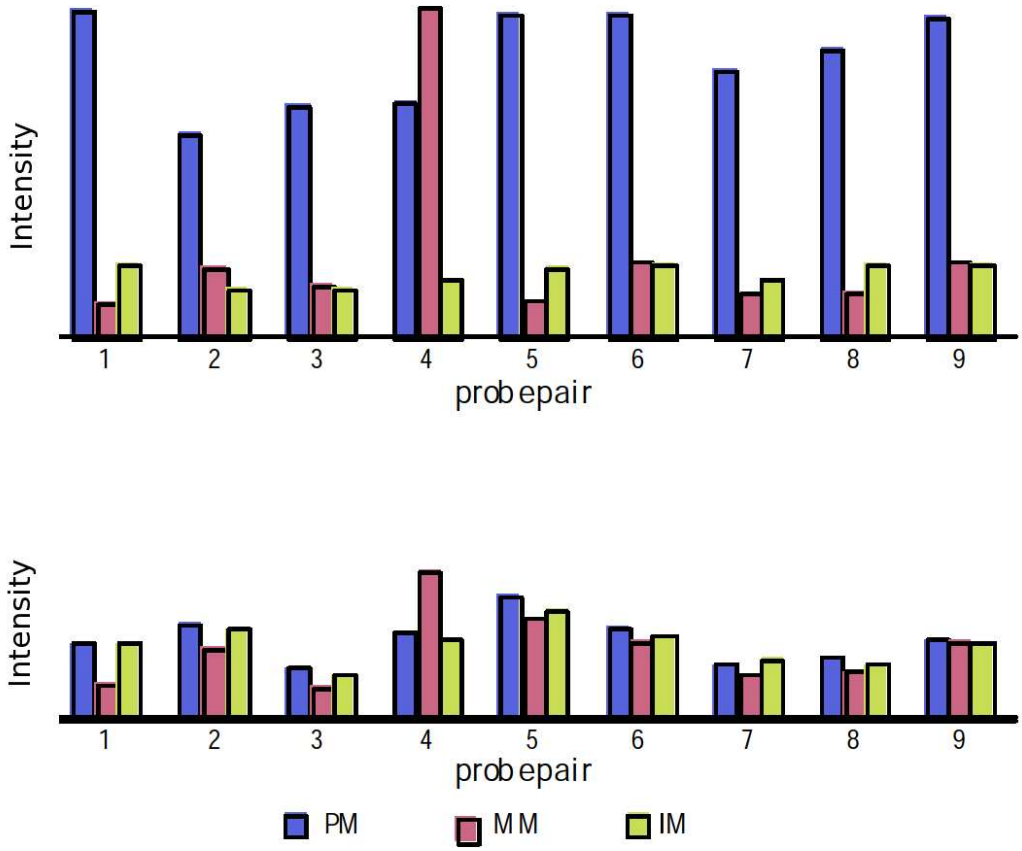

Figure 5.7: Comparison of IM and MM for a hypothetical dataset (Fig. reproduced from [7]). In the top panel, most of the MM values are smaller than PMs, i.e. $MM_j = IM_j$ for $j = 1, \ldots 9,\ j \neq 4$. The yellow bars indicate the estimated IM value which would be used if $MM > PM$. For probe pair 4, the MM_4 is larger than the PM_4, so it is not a useful value for estimating the stray signal component of PM_4. An imperfect, but useful resolution is to estimate a MM_4 value that is typical for the probe set. For the overall probe set, the SB_i is large, so it can be used directly to estimate an IM_4 value for MM_4. In the second panel, SB_i is small, so an accurate estimate for MM_4 can not be computed from it. The best estimate is a IM_4 slightly smaller than the PM_4.

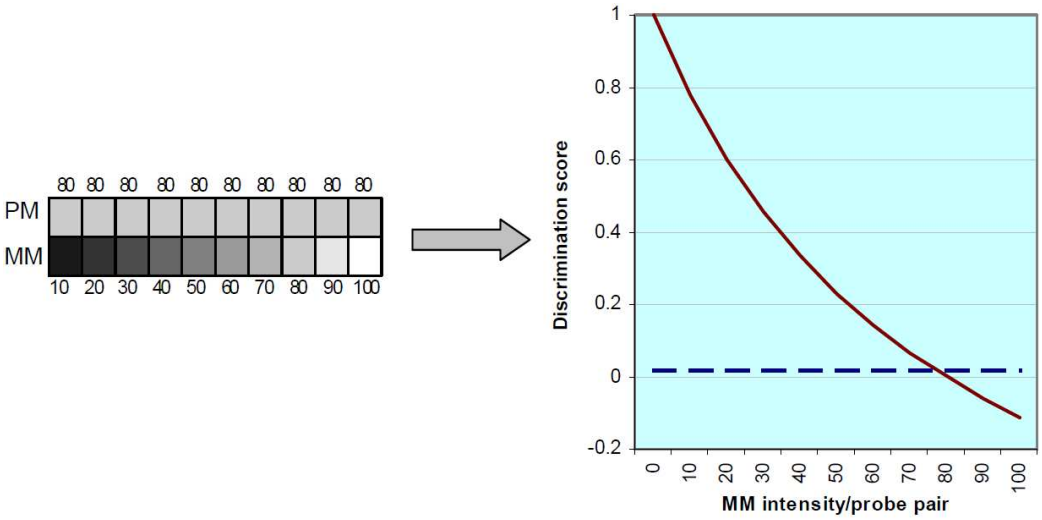

Figure 5.8: Discrimination score (figure reproduced from [5]). Left: hypothetical probe set consisting of 10 probe pairs. While the PM value is set to 80 for each probe pair the MM values increase from 10 to 100. Right: the discrimiation score calculated for the 10 probe pairs. As the MM intensity increases the discrimination score decreases. The dashed line shows the user defineable parameter τ (default $\tau = 0.015$.)

Chapter 6

BSS and Microarray Data

In this chapter it is shown how the data obtained from microarray experiments fits into the linear mixture model

$$\underline{\mathbf{X}} = \underline{\mathbf{A}}\ \underline{\mathbf{S}}. \tag{6.1}$$

However, also the limits of this model will be discussed in the context of the various gene regulation mechanisms occurring in prokaryotic and eukaryotic cells. Based on this considerations it will be explained in which microarray experiments BSS based algorithms may lead to fruitful results and where these methods are prone to fail.

6.1 Linear mixture model and microarray data

Before BSS techniques can be applied to microarray data it must be clarified how the observed gene expressions can be explained by means of the linear mixture model (6.1). In other words, the meaning of the matrices $\underline{\mathbf{A}}$ and $\underline{\mathbf{S}}$

has to be explained in a biological context. For this purpose, the following assumptions are usually made [53]:

1. The various biological processes occurring in a cell do not interact with each other.

2. Each of the individual processes leads to the expression of a specific set of genes. Thereby, some genes my be expressed by several cellular processes.

3. The absolute expression levels of the individual genes participating in the same cellular process may vary.

4. Cells react to external or internal stimuli by up- or downregulating their individual biological processes. This leads to the joint up- or downregulation of the participating genes.

5. The gene expression levels recorded by a microarray are the composition of the individual expression levels caused by the various cellular processes occurring in the cell.

Now assume that the cells under investigation were exposed to M various stimuli and that for each of these stimuli a microarray was carried out. Furthermore, assume that the microarray is capable of detecting the expression of T different genes in parallel. In such a case, the obtained data is used to constitute the $M \times T$ matrix $\underline{\mathbf{X}}$, i.e. the element x_{mt} contains the expression of the t-th gene under the m-th environmental condition. Based the assumptions made above the n-th row of the $N \times T$ matrix $\underline{\mathbf{S}}$ is then considered to contain the expression levels of the n-th cellular process, i.e. the element

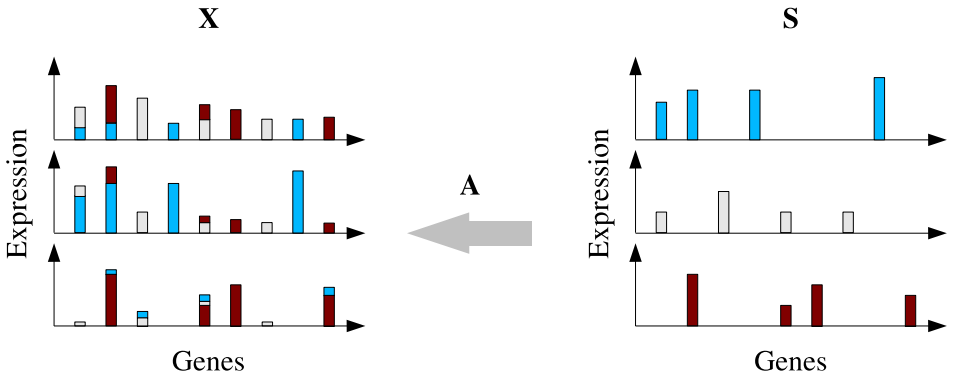

Figure 6.1: Linear mixture model applied to microarray data. Right: gene expression pattern of three cellular processes. These patterns constitute the rows of the matrix $\underline{\mathbf{S}}$. Left: the observed expression levels are assumed to be the weighted sums of the expression levels of the three underlying cellular processes. Depending on the external stimuli these processes are up or downregulated. The corresponding weights are stored in the matrix $\underline{\mathbf{A}}$. Note, that the absolute expression levels of the individual genes participating in the same cellular process my vary while they are jointly up- or downregulated under the various stimuli.

s_{nt} contains the expression of the t-th gene by the n-th cellular process. The activity of this process under the m-th stimuli is encoded into the element a_{mn} of the mixing matrix $\underline{\mathbf{A}}$. In other words, the elements of the n-th column of $\underline{\mathbf{A}}$ reflect, if the n-th biological process was up- or downregulated under various stimuli (see Fig. 6.1).

6.2 Limits of the linear mixture model

Even if at first glance the linear mixture model presented in the last section seems to be adequate to explain cellular processes it still ignores two important aspects of gene regulation. On the one hand the expression levels of many genes are often jointly determined by several transcription factors

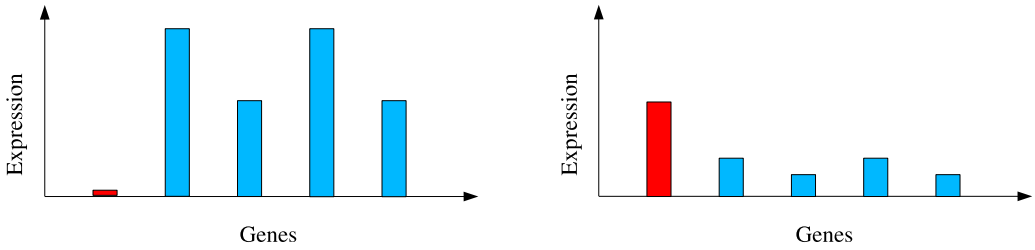

Figure 6.2: Negative transcription factors

and negative transcription factors exist (cf. Sec. 4.1.3). On the other end, some cellular processes react to external stimuli by alternative splicing [55]. As will be discussed in the sequel, both mechanisms can not be incorporated into the linear mixture model.

6.2.1 Transcription Factors and BSS

The goal of the BSS analysis of microarray data is to determine which genes participate at a particular cellular process. Accordingly, also the transcription factors which steer the expression of the individual genes should be grouped together with the other genes, i.e. should appear in the same row of $\underline{\mathbf{S}}$. However, such transcription factors can not be modelled by the linear mixture model if they suppress the expression of other genes. This is illustrated in Fig. 6.2. The depicted genes are supposed to belong to the same biological process. Thereby, the first gene (red bar) is assumed to be a negative acting transcription factor which, when expressed, hampers the expression of the other genes. If the transcription factor is downregulated, the remaining genes are expressed efficiently (Fig. 6.2 left). Hence, the corresponding entry in the mixing matrix should be large to allow for these genes but should simultaneously be small in order to allow for the low expression

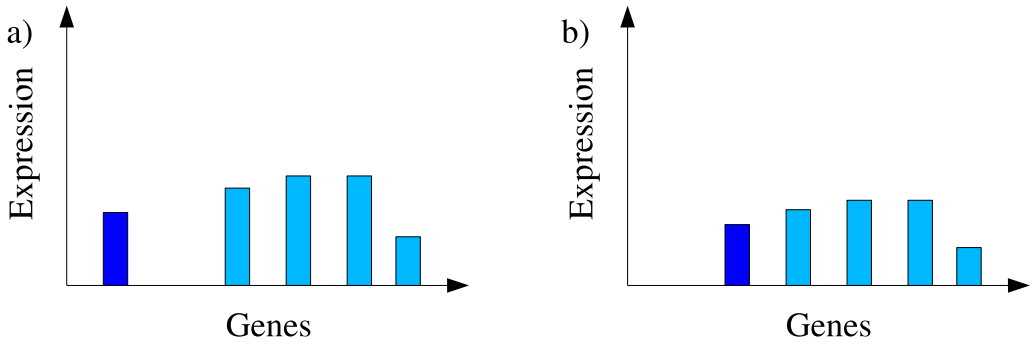

Figure 6.3: Problems occurring when several transcription factors regulate the same genes.

level of the negative transcription factor. Similarly, if the negative transcription factor is highly expressed the remaining genes are downregulated (Fig. 6.2 right). Thus, the corresponding element in the mixing matrix should be large in order to reflect the high expression of the negative transcription factor and should simultaneously be small in order to account for the repression of the other genes. However, negative transcription are rather rare in higher eukaryotes such that this phenomena is not likely to appear very often.

Similar problems arise when genes are jointly regulated by more than just one transcription factor (like, e.g. in the regulation of the Endo16 gene discussed in 4.1.3.2). To illustrate this consider Fig. 6.3. There, two positive transcription factors (dark blue) are used to regulate the remaining genes of the cellular process. Each of these transcription factors can lead to an expression of the remaining genes on its own, i.e. increased expression of only the first or only the second transcription factor also leads to a proportional increase in the expression the remaining genes (Fig. 6.3 a) and b)). Hence, even if both transcription factors belong to the same biological process, they still may not appear in the same row of $\underline{\mathbf{S}}$, as this would mean that both of

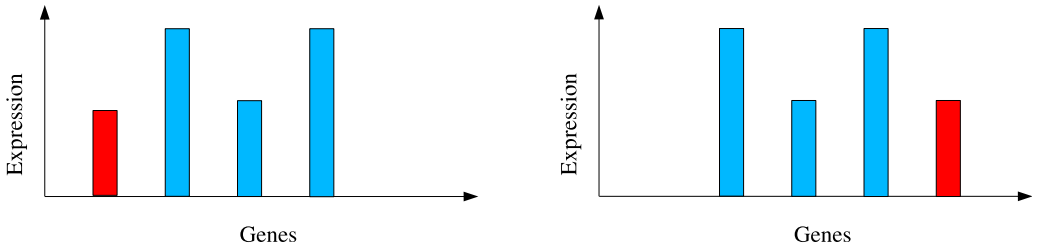

Figure 6.4: Reaction to external stimuli by alternative splicing.

them are necessary for the expression of the remaining genes.

6.2.2 Alternative Splicing and BSS

Recent research has shown [55] that cells also react to external stimuli by alternative splicing. This means, that some of the genes belonging to a particular cellular process are spliced differently when they are exposed to stress. Such a scenario is illustrated in Fig. 6.4 for two different stimuli. It is supposed, that the first splicing form (corresponding to the red bar in the left of Fig. 6.4) is only expressed under a certain stimuli while under a different stimuli another splicing form is used (corresponding to the red bar in the right of Fig. 6.4). Obviously, such a behaviour cannot be modelled by the linear mixture model assumed above.

6.3 Suitability of the BSS model

Alternative splicing but also the regulation of genes by several transcription factors obviously lead to violations of the linear mixture model assumed in BSS. However, it must be noted that the role of alternative splicing in the context of varying stimuli is not fully understood yet. Hence, it is hard to

estimate if alternative splicing should be considered to be an exception rather than the rule.

In contrast, it is commonly accepted that many genes are jointly regulated by several transcription factors. However, it also turns out that genes, which are expressed by cells to cope with external stress (like, e.g. heat) are often steered by a single transcription factor only. The reason for this seems to be that cells have to react rapidly to such stresses in order to avoid being seriously damaged. Hence, only single transcription factors should be needed to trigger stress reaction processes.

In contrast, genes which are only needed at certain stages of the cell cycle or at certain points during the development of their host organism are often controlled by a multitude of transcription factors (cf. Endo16 in Sec. 4.1.3). This eventually means, that BSS analysis techniques should preferably be applied to experiments, where the cells were exposed to various stimuli or stress while they seem to be less suited to analyse data sets which were acquired at different points of the cell cycle or at different stages of the development of the organism.

In practice, genes which do not follow the linear mixture model must be considered as noise or outliers such hampering the applied BSS algorithms to find meaningful results. As we have observed, this effect is particularly striking when BSS algorithms are used which do not assume nonnegativity of the matrices $\underline{\mathbf{A}}$ and $\underline{\mathbf{S}}$, like e.g. ICA algorithms. In such cases, sources (i.e. rows of $\underline{\mathbf{S}}$) are obtained which simultaneously contain positive as well as negative entries. Seen from a biological point of view this would been that some genes have been expressed negatively or, in other words, that negative

amounts of mRNA have been transcribed. Obviously, it is difficult in such cases to interpret the data at all.

Of course, also NMF algorithms are affected by violatons of the linear mixture model. Still, they always lead to nonnegative sources thus facilitation the interpretation of the data.

Chapter 7

Novel NMF approaches

As pointed out at the end of the last chapter, ICA methods have the decisive drawback that they lead negative entries in the estimated sources when the linear mixture model is violated. As these negative entries are not interpretable in the context of microarray data matrix factorisation techniques are needed which lead, on the one hand, to nonnegative matrices $\underline{\mathbf{A}}$ and $\underline{\mathbf{S}}$ but which must also be capable to produce unique results (up to some scaling and permutations). For this purpose, extensions of the NMF algorithm with sparseness constraints as well as of the standard NMF algorithm are presented in this chapter. The suitability of these algorithms to solve the BSS problem will be tested thoroughly by means of several simulations. Finally, a new algorithm, called sparse NMF, will be applied to real world microarray data.

7.1 Extended Sparse NMF

While constraining all rows of $\underline{\mathbf{S}}$ to have a common sparseness improves the results of NMF in image decomposition applications significantly [41], more general constraints are needed in BSS, where the sources to be recovered may have all different sparseness. To cope with this problem, we have introduced an extended sparse NMF (esNMF) algorithm in [86] (cf. Alg. 1) in which the rows of $\underline{\mathbf{S}}$ may have different sparseness.

esNMF is an extension of sparse NMF algorithm derived by Hoyer (cf. Sec. 3.2), in which individual sparseness constraints σ_k, $k = 1 \ldots n$, for each of the rows of $\underline{\mathbf{S}}$ may be provided while no assumptions concerning the sparseness of $\underline{\mathbf{A}}$ are made. Note, that the update steps of the matrices $\underline{\mathbf{A}}$ and $\underline{\mathbf{S}}$ in Alg. 1 (steps 3 and 6) are the same as in Sec. 3.2.

We observed that the devised NMF algorithm is very dependent on the order in which differing sparsenesses are presented if the latter are chosen differently for the different sources. It turns out that the initialisation of the matrices $\underline{\mathbf{A}}$ and $\underline{\mathbf{S}}$ determine the order in which the sources will be recovered. If this order differs from the order in which the sparsenesses are supplied within the algorithm, poor results can be expected only. This fact is accounted for in the algorithm by the special way matrix $\underline{\mathbf{S}}$ is initialised in step 2 as well as the adaptive assignment of the sparseness constraints of the columns of $\underline{\mathbf{S}}$ in steps 4 and 5.

We try to circumvent this problem by initialising the rows of $\underline{\mathbf{S}}$ such that they are all of equal sparsity and unbiased towards the original sources. The latter condition can be met by demanding that the rows of $\underline{\mathbf{S}}$ have a much

Input: observation data matrix $\underline{\mathbf{X}}$, sparseness constraints σ_k, $k = 1, \ldots, n$, (assume $\sigma_1 \leq \sigma_2 \cdots \leq \sigma_n$ for simplicity)

Output: decomposition $\underline{\mathbf{XS}}$ of $\underline{\mathbf{X}}$ such that rows of $\underline{\mathbf{S}}$ fulfil given sparseness constraints σ_k, $k = 1 \ldots n$

1 Initialise $\underline{\mathbf{A}}$ to a random non-negative matrix.
2 Initialise $\underline{\mathbf{S}}$ to a random non-negative matrix where all rows have sparseness 0.9.
repeat
3 Set $\underline{\mathbf{S}} \leftarrow \underline{\mathbf{S}} - \mu_S \underline{\mathbf{A}}^T(\underline{\mathbf{A}}\ \underline{\mathbf{S}} - \underline{\mathbf{X}})$.
4 Determine the current sparseness $\tilde{\sigma}_i$, $i = 1, ..., n$, of each row of $\underline{\mathbf{S}}$. Find permutation $\{j_1, \ldots, j_n\}$ of $\{1, \ldots, n\}$ such that $\tilde{\sigma}_{j_1} \leq \tilde{\sigma}_{j_2} \leq \cdots \leq \tilde{\sigma}_{j_n}$.
5 Project the j_k-th row of $\underline{\mathbf{S}}$ such that it fulfils the sparseness constraint σ_k, $k = 1, ..., n$ (cf. Alg. 2).
6 Set $\underline{\mathbf{A}} \leftarrow \underline{\mathbf{A}} \otimes (\underline{\mathbf{X}}\ \underline{\mathbf{S}}^T) \oslash (\underline{\mathbf{A}}\ \underline{\mathbf{S}}\ \underline{\mathbf{S}}^T)$.
until *convergence*;
Note: in the last step $\otimes$ and $\oslash$ symbolise elementwise multiplication and division respectively.

Algorithm 1: Extended Sparse NMF Algorithm

higher sparseness than the original sources. In our experiments, we always used an initial sparseness of 0.9 for the rows of $\underline{\mathbf{S}}$.

During the first iteration $\underline{\mathbf{S}}$ is updated under the assumption that the randomly initialised matrix $\underline{\mathbf{A}}$ is the actual mixing matrix. This leads to a first estimate of the original sources of which the sparsenesses can be computed. Even if this estimate is still very rough a comparison between the original and the estimated sparsenesses (cf. step 4 in Alg. 1) can already reveal the order in which the sources will be recovered. Hence, we can now project the estimated sources such that they fulfil their corresponding sparseness constraints (step 5) using the projection operator as described in Sec. 7.1.1, Alg. 2. Finally, in step 6 of esNMF, the estimated mixing matrix $\underline{\mathbf{A}}$ is updated under the assumption that $\underline{\mathbf{S}}$ contains the actual sources.

7.1.1 Sparse projection

In [86] we have discussed the uniqueness of the sparse projection operator defined in [41]. The major points of this prove are as follows:

Basically the sparse NMF algorithm uses a projection step as follows: Given $\mathbf{x} \in \mathbb{R}^n$ and fixed $\lambda_1, \lambda_2 > 0$, find $\mathbf{s}$ such that

$$\mathbf{s} = \text{argmin}_{\|s\|_1=\lambda_1, \|s\|_2=\lambda_2, s\geq 0} \|\mathbf{x} - \mathbf{s}\|_2. \tag{7.1}$$

We want to solve problem (7.1) by projecting $\mathbf{x}$ onto

$$M := \{\mathbf{s} | \|\mathbf{s}\|_1 = \lambda_1\} \cap \{\mathbf{s} | \|\mathbf{s}\|_2 = \lambda_2\} \cap \{\mathbf{s} \geq 0\}. \tag{7.2}$$

In order to solve equation (7.1), $\mathbf{x}$ has to be projected onto a point adjacent to it in M:

Definition 3. *A point* $\mathbf{p} \in M \subset \mathbb{R}^n$ *is called* adjacent to $\mathbf{x} \in \mathbb{R}^n$ in M, *in symbols* $\mathbf{p} \triangleright_M \mathbf{x}$ *or shorter* $\mathbf{p} \triangleright \mathbf{x}$, *if* $\|\mathbf{x}-\mathbf{p}\|_2 \leq \|\mathbf{x}-\mathbf{q}\|_2$ *for all* $\mathbf{q} \in M$.

In the following we will study in which this is possibly, and in which cases this projection is even unique.

7.1.1.1 Indeterminacies

First we will discuss the question of existence of projection points $\mathbf{p}$. Obviously $\mathbf{p}$ cannot exist if $\mathbf{x}$ lies in the closure of M (i.e. 'touches' it) without being an element of M. Indeed this is the only obstruction to existence as the following remark shows. Proofs are not given due to space limitations.

Remark 1 (Existence). *If* M *is closed and nonempty, then for every* $\mathbf{x} \in \mathbb{R}^n$ *there exists* $\mathbf{p} \in M$ *with* $\mathbf{p} \triangleright \mathbf{x}$.

In order to study uniqueness of the projection, we define an exception set as follows:

Definition 4. *Let* $\mathcal{X}(M) := \{\mathbf{x} \in \mathbb{R}^n |$ *there exists more than one point adjacent to* $\mathbf{x}$ *w.r.t.* $M\} = \{\mathbf{x} \in \mathbb{R}^n | \#\{\mathbf{p} \in M | \mathbf{p} \triangleright \mathbf{x}\} > 1\}$ *denote the* exception set *of* M.

So the exception set contains the set of points from which no unique projection is possible. We want to show that this set vanishes or is at least very small. Obviously the exception set of an affine linear hyperspace vanishes. Indeed, we can prove more generally that the exception set of a convex set

is empty. In general however, we cannot expect $\mathcal{X}(M)$ to vanish altogether. However we can show that in practical applications we can easily neglect it:

Theorem 2 (Uniqueness). *vol*$(\mathcal{X}(M)) = 0$.

This theorem states that the Lebesgue measure of the exception set is zero i.e. that it does not contain any open ball. In other words, if $\mathbf{x}$ is drawn from a continuous probability distribution on $\mathbb{R}^n$, then $\mathbf{x} \in \mathcal{X}(M)$ with probability 0. In practice, this proves uniqueness because samples are usually drawn from continuous distributions.

The theorem follows from the fact that if $\mathbf{x}$ is some point of the exception set of M, any point lying on a line between $\mathbf{x}$ and one of its projections does not again lie in the exception set.

7.1.1.2 Projection algorithm

Now let M be defined by equation 7.2. Hoyer [41] essentially proposes algorithm 2 to project a given vector $\mathbf{x}$ onto $\mathbf{p} \in M$ such that $\mathbf{p} \triangleright \mathbf{x}$. The algorithm iteratively detects $\mathbf{p}$ by first satisfying the L_1-norm condition and then the L_2-norm condition. It terminates if the constructed vector is already positive; otherwise a negative coordinate is selected, set to zero and the search is continued in $\mathbb{R}^{n-1}$.

The algorithm terminates after maximally $n - 1$ iterations. Indeed, it finds the correct projections if $\mathbf{x} \notin \mathcal{X}(M)$ as the following theorem shows. It is proved by showing that in each step the new estimate has $\mathbf{p}$ as closest point in M.

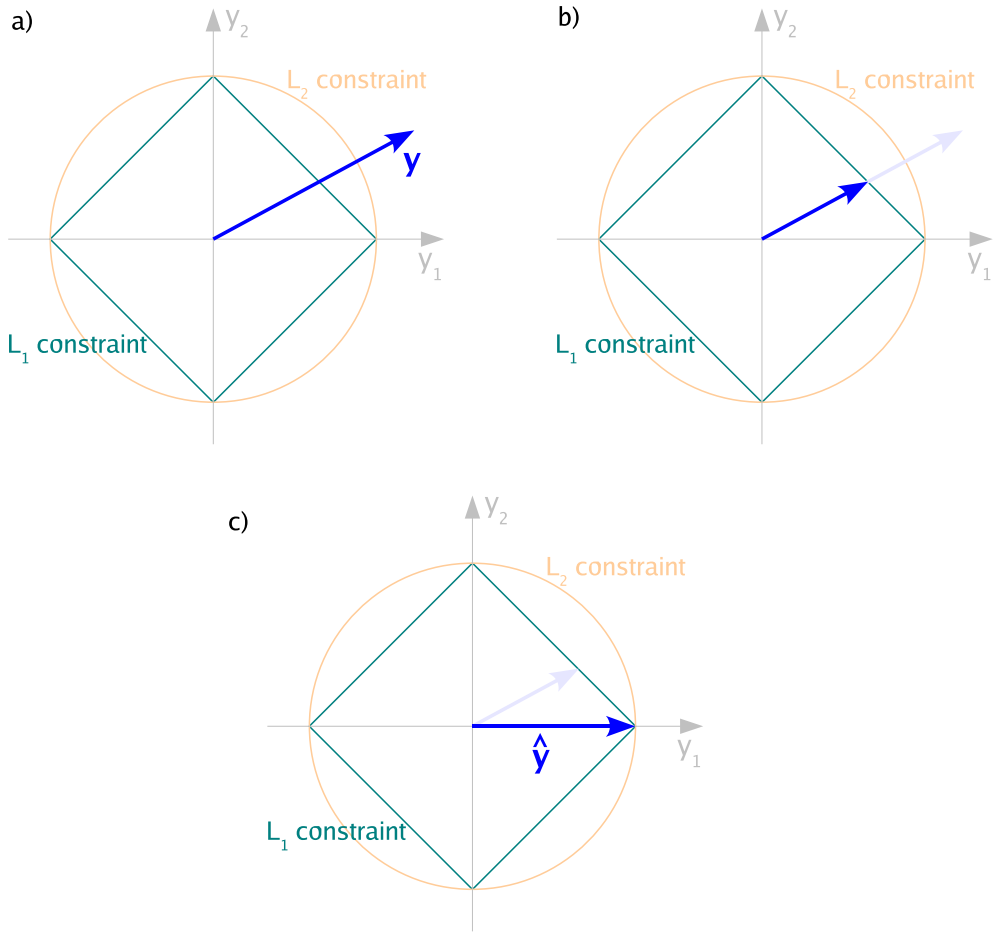

Figure 7.1: Projection operator for the 2D case. Left: The vector $\mathbf{y}$ is to be projected such that it fulfils the given L_1 and L_2 norm constraints. Right: After the first projection step the vector has the desired L_1 norm. Bottom: After the second step also the L_2 norm constraint is fulfilled.

Input: vector $\mathbf{x} \in \mathbb{R}^n$, norm conditions λ_1 and λ_2
Output: closest non-negative $\mathbf{s}$ with $\|\mathbf{s}\|_i = \lambda_i$

repeat
1 Set $\mathbf{r} \leftarrow \mathbf{x} + (\|\mathbf{x}\|_1 - \lambda_1/n)\mathbf{e}$ with $\mathbf{e} = (1, \ldots, 1)^\top \in \mathbb{R}^n$.
2 Set $\mathbf{m} \leftarrow (\lambda_1/n)\mathbf{e}$.
3 Set $\mathbf{s} \leftarrow \mathbf{m} + \alpha(\mathbf{r} - \mathbf{m})$ with $\alpha > 0$ such that $\|\mathbf{s}\|_2 = \lambda_2$.
if *exists* j *with* $s_j < 0$ **then**
4 Fix $s_j \leftarrow 0$.
5 Remove j-th coordinate of $\mathbf{x}$.
6 Decrease dimension $n \leftarrow n - 1$.
7 **goto** 1.
end
until $\mathbf{s} \geq 0$;

Algorithm 2: Sparse projection

Theorem 3 (Sparse projection). *Given* $\mathbf{x} \geq 0$ *such that* $\mathbf{x} \notin \mathcal{X}(M)$ *and let* $\mathbf{p} \in M$, $\mathbf{p} \triangleright \mathbf{x}$ *be its projection onto* M. *If* $\mathbf{s}$ *is constructed by line 3 of algorithm 2, then* $\mathbf{p} \triangleright \mathbf{s}$ *and* $\mathbf{s} \notin \mathcal{X}(M)$. *Furthermore, the algorithm terminates at* $\mathbf{s} = \mathbf{p}$.

7.1.2 Simulations

7.1.3 Adaptive sparseness assignment

In this section we show that the initialisation of the matrix $\underline{\mathbf{A}}$ actually determines the order in which the sources are estimated, and demonstrate that the esNMF algorithm may automatically detect this order. Furthermore, it will be shown that this automatic detection is pivotal for the esNMF algorithm to find good solutions independent of the order in which the given sparsenesses are supplied to it.

For this purpose we compared our esNMF algorithm with a second, sim-

plified version of it, called esNMF$_{fix}$, in which the matrix $\underline{\mathbf{S}}$ is initialised by a random nonnegative matrix and in which the given sparseness constraints are applied to the rows of $\underline{\mathbf{S}}$ in the fixed order in which they were supplied to the algorithm.

We evaluated the performance of both algorithms by applying them to a toy BSS problem in which three nonnegative sources with different sparsenesses should be recovered from three observations. The three sources s_i^{org}, $i = 1, \ldots, 3$, consisted of 3000 data points and had sparsenesses of $\sigma_1 = 0.61$, $\sigma_2 = 0.47$ and $\sigma_3 = 0.23$ respectively. The observation matrix $\underline{\mathbf{X}}$ was generated by multiplying the source matrix $\underline{\mathbf{S}}^{org}$ by a random nonnegative 3×3 matrix $\underline{\mathbf{A}}^{org}$ as in (**??**).

In this case, there are 6 possible orders in which the given sparsenesses σ_i, $i = 1, \ldots, 3$, can be provided to the algorithms. For each of these orders both algorithms were used to recover the original sources s_i^{org} as well as the mixing matrix $\underline{\mathbf{A}}^{org}$. As we wanted to show that the initialisation of the matrix $\underline{\mathbf{A}}$ determines the order in which the sources are recovered, we initialised it in both algorithms by the original mixing matrix $\underline{\mathbf{A}}^{org}$. Furthermore, as both algorithms are known to be prone to local minima, the algorithms were run 100 times for each sparseness order, each time with a different initialisation of the matrix $\underline{\mathbf{S}}$.

The results of both algorithms were compared using the crosstalking error (CTE) between the original and the estimated mixing matrix as well as by the correlation coefficients (CC) between the original and the recovered sources.

As can be seen in Figures 7.3 and 7.2, the esNMF$_{fix}$ algorithm only succeeded in recovering the original sources and the mixing matrix when the

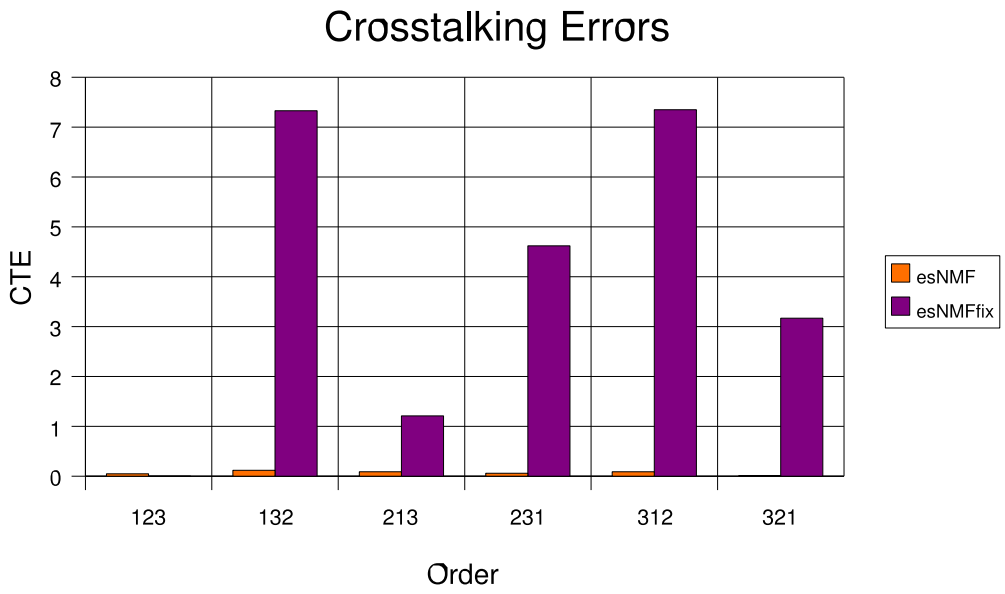

Figure 7.2: Comparison of the esNMF$_{fix}$ and the esNMF algorithm by means of the CTE of the recovered mixing matrix. Order indicates the order in which the individual sparsenesses were provided to the algorithms. The depicted CTE is the average of the 100 CTE's computed in the simulations. Obviously, esNMF leads to good results regardless of the order of the sparsenesses.

sparsenesses were provided in the same order as the original sources. If we initialised the matrix $\underline{\mathbf{A}}$ in esNMF$_{fix}$ by a matrix $\underline{\tilde{\mathbf{A}}}$ which was equal to $\underline{\mathbf{A}}$ up to a permutation of two of its columns, then the order of the corresponding sparsenesses had to be permuted as well in order to obtain satisfying results. Hence, we conclude that the initialisation of the matrix $\underline{\mathbf{A}}$ actually determines the order in which the sources are recovered.

On the other hand the esNMF algorithm only failed in maximally 10% of its runs regardless of the order in which the sparsenesses were provided. Apart from these failures, it always lead to CTE's smaller then 10^{-7} as well as to CCs higher than 0.9999. This confirms the eligibility of the proposed

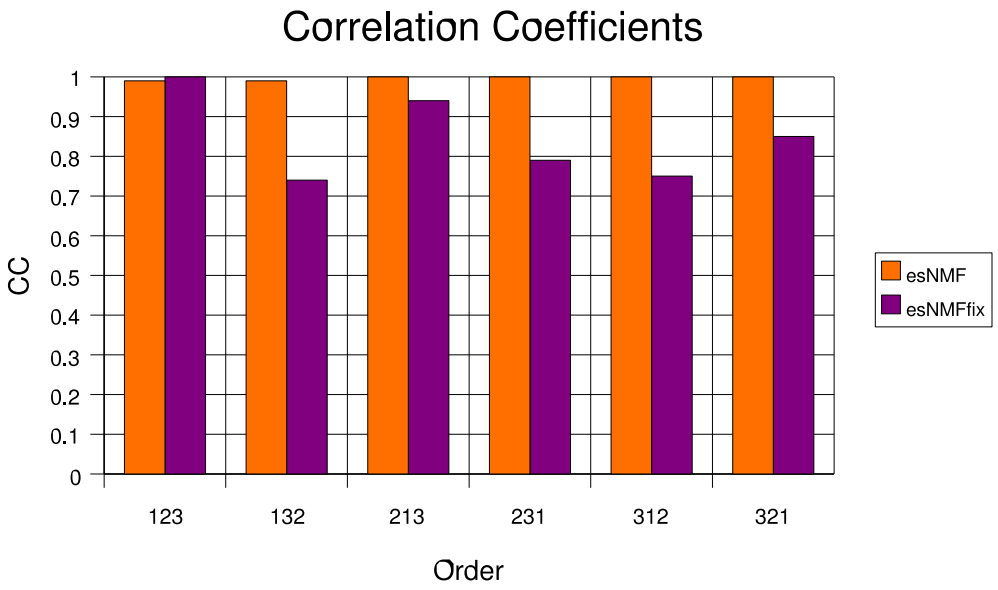

Figure 7.3: Comparison of the esNMF$_{fix}$ and the esNMF algorithm by means of the original and the recovered sources. Order indicates the order in which the individual sparsenesses were provided to the algorithms. The depicted CCs are the average of the 100 CC's computed in the simulations. Obviously, esNMF leads to good results regardless of the order of the sparsenesses.

adaptive assignment method.

7.1.3.1 Sparseness Presets and Robustness

In real life experiments it is usually difficult to estimate the sparsenesses of the sources a priori. Accordingly, we have tested the esNMF algorithm for its robustness against sparseness presets $\hat{\sigma}_k$ which deviated from the actual sparsenesses σ_k of the sources.

For the simulations we used three nonnegative sources with sparsenesses of $\sigma_1 = 0.61$, $\sigma_2 = 0.47$ and $\sigma_3 = 0.23$ respectively and a nonnegative 3×3 mixing matrix $\underline{\mathbf{A}}^{org}$. Both the sources and the mixing matrix could be recovered perfectly (CC > 0.9999, CTE $< 1.8 \cdot 10^{-7}$) if the correct sparsenesses

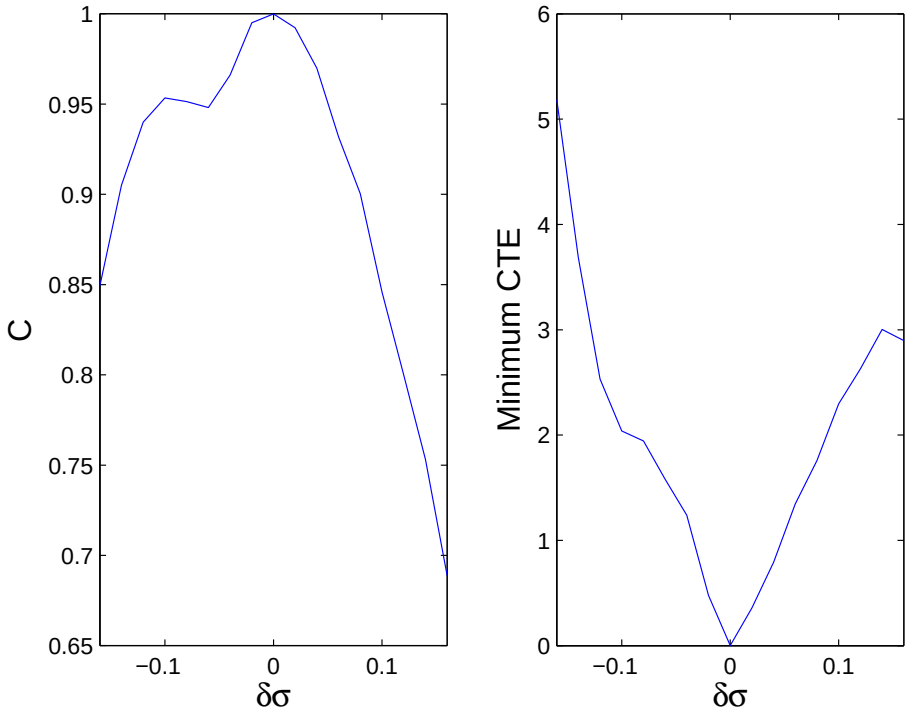

Figure 7.4: The dependence of C (see text) and the CTE on the sparseness presets.

were provided to the esNMF algorithm.

Then, we shifted the sparseness presets $\hat{\sigma}_k$ from their correct values σ_k by adding an error $\delta\sigma$ ($\hat{\sigma}_k = \sigma_k + \delta\sigma$, $k = 1, \ldots, n$), where $\delta\sigma$ was varied from -0.16 to 0.16 in steps of 0.02. For each value of $\delta\sigma$ the esNMF algorithm was run 20 times with different initialisations for the matrices $\underline{\mathbf{A}}$ and $\underline{\mathbf{S}}$. Now, for each of the 20 runs the minimal correlation coefficient $c_j^{\min}(\delta\sigma), j = 1, \ldots, 20$, between the estimated sources and their original counterparts was determined. Furthermore, the CTEs between the estimated and the original mixing matrices were computed. The minimum of these 20 CTEs as well as

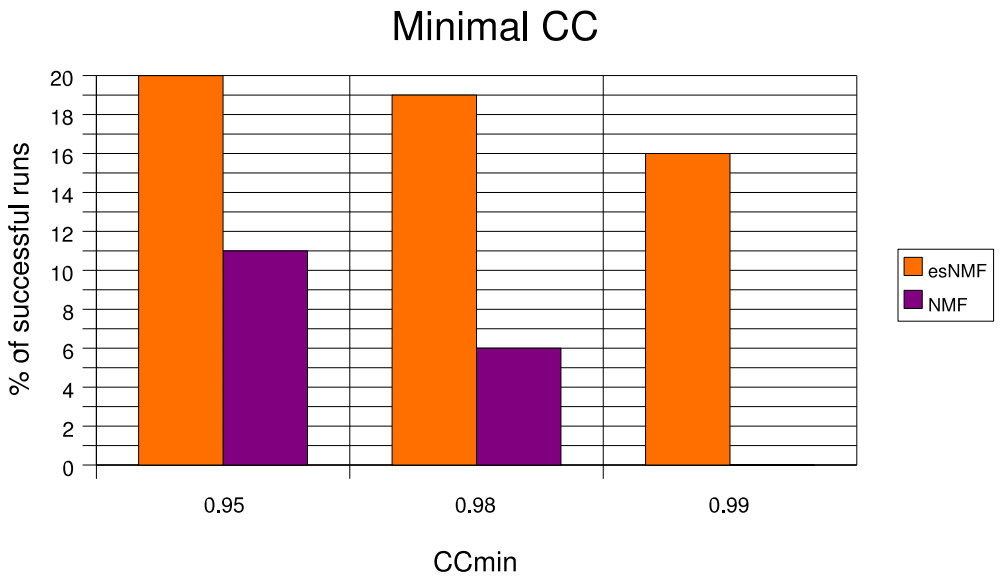

Figure 7.5: Comparison NMF with esNMF. Obviously, esNMF leads much more often to successfull results than standard NMF.

the maximum value C of $c_j^{\min}(\delta\sigma), j = 1, \ldots, 20$, are depicted in Fig. 7.4 for each $\delta\sigma$.

While the CTE seemed to be slightly more sensitive to too small presets, the sources could still be recovered well down to a $\delta\sigma$ of about -0.1. Hence, we would suggest to use rather too low than too high presets for the sparsenesses in real life applications. Generally, a $\delta\sigma$ between -0.5 and 0.5 should still lead to acceptable results.

7.1.4 Comparison NMF - esNMF

The devised esNMF algorithm was also compared to the standard NMF algorithm. For this purpose 100 random nonnegative 3×1000 source matrices $\underline{\mathbf{S}}$ were created. Thereby, the sparseness of the individual rows of $\underline{\mathbf{S}}$ was fixed to

0.61, 0.47 and 0.23 respectively. Each of these source matrices was multiplied by another nonnegative random 3×3 mixing matrix $\underline{\mathbf{A}}$ such that eventually 100 observation matrices $\underline{\mathbf{X}}$ were obtained. These matrices were provided to NMF as well as to esNMF and the results were compared by means of the correlation coefficients between the original and the corresponding estimated sources. The BSS problem was considered to be solved successfully, if the minimal correlation coefficient between the estimated and the corresponding original sources was larger than a threshold CC_{min}. The results for various values of CC_{min} are depicted in Fig. 7.5. As can be seen, taking into account preknowledge in the form of sparseness constraints esNMF leads much more often to successfull recoveries of the sources that NMF. Furthermore, perfect recoveries of the sources ($CC_{min} > 0.99$) can only be obtained by esNMF.

7.1.5 Summary of esNMF

In this section a novel extension of Hoyers NMF with sparseness constraints algorithm, called extended sparsen NMF, was presented. In this algorithm preknowledge in the form of the sparseness of the sources can be incorporated. As was shown, this additional knowledge helps to improve the reliability and quality of the obtained results in comparison to standard NMF methods. Thereby, the preknowledge about the sparsenesses of the individual sources needs not to be known precisely, but generally, rather too small than too large sparseness presets should be used. The devised algorithm leads to the desired nonnegative factorisation of the matrix $\underline{\mathbf{X}}$, however, it can only be applied to microarray data where the number of genes participating at the

individual processes is known or can be approximated *a priori*.

7.2 Sparse Nonnegative Matrix Factorisation

As was already outlined in Sec. 3 NMF is a powerful tool to analyse nonnegative data matrices. However, the nonnegativity constraints made in NMF are insufficient to obtain a unique decompositions of the data matrix. Hence, NMF cannot be applied to BSS problems if no further assumptions are made.

For this reason, an extension of NMF, called sparse NMF (sNMF), is presented in this chapter in which sparseness assumptions on the rows of $\underline{\mathbf{S}}$ are made. As will be shown in simulations, this additional assumption is sufficient to solve the BSS problem up to the usual scaling and permutation indeterminacies. As the target function of the devised algorithm contains many local minima, a genetic algorithm (GA) will be used for its optimisation. This GA will be designed such that it can be run on large clusters of computers thus decreasing computation times drastically.

7.2.1 Definition of sNMF

In order to obtain unique results, the nonnegativity constraints made in NMF are extended by a sparseness a assumption on the rows of $\underline{\mathbf{S}}$ in sNMF. The goal of sNMF is then to factorise $\underline{\mathbf{X}}$ by two nonnegative matrices $\underline{\mathbf{A}}$ and $\underline{\mathbf{S}}$ such that the rows of $\underline{\mathbf{S}}$ are as sparse as possible. Formally, this leads to the following definition of sNMF:

Definition 5. *Sparse Nonnegative Matrix Factorisation (sNMF)*
Given a nonnegative $M \times T$ matrix $\underline{\mathbf{X}}$ find two nonnegative matrices $\underline{\mathbf{A}}$ and

$\underline{\mathbf{S}}$ *of size* $M \times N$ *and* $N \times T$ *respectively such that*

$$F(\underline{\mathbf{A}}, \underline{\mathbf{S}}) = \|\underline{\mathbf{X}} - \underline{\mathbf{A}}\ \underline{\mathbf{S}}\|^2 \tag{7.3}$$

is minimised whereas

1. $\underline{\mathbf{A}}$ *and* $\underline{\mathbf{S}}$ *are nonnegative,*

2. *the rows of* $\underline{\mathbf{S}}$ *contain as many nil entries as possible.*

This problem can be solved by means of the following objective function which has to be minimised with respect to $\underline{\mathbf{A}}$ and $\underline{\mathbf{S}}$:

$$E(\underline{\mathbf{A}}, \underline{\mathbf{S}}) = \underbrace{\frac{\|n(\underline{\mathbf{X}}) - n(\underline{\mathbf{A}}\ \underline{\mathbf{S}})\|^2}{2M}}_{factorisation\ term} - \lambda \underbrace{\frac{1}{N} \sum_{n=1}^{N} \sigma(\tilde{\mathbf{s}}_n)}_{sparseness\ term} . \tag{7.4}$$

Here, the function n is used to normalise the rows of $\underline{\mathbf{X}}$ and $\underline{\mathbf{A}}\ \underline{\mathbf{S}}$, i.e.

$$\begin{aligned} n : \mathbb{R}^{M \times T} &\to \mathbb{R}^{M \times T} \\ n(\underline{\mathbf{X}}) &\mapsto \underline{\tilde{\mathbf{X}}} \quad \text{where} \quad \tilde{\mathbf{x}}_{ij} = \frac{x_{ij}}{\sqrt{\sum_{k=1}^{T} x_{mk}^2}} \end{aligned} \tag{7.5}$$

Hence, $\|n(\underline{\mathbf{X}}) - n(\underline{\mathbf{A}}\ \underline{\mathbf{S}})\|^2$ may maximally become $2M$ such that division by this factor normalises the factorisation term in (7.4), i.e. $0 \leq \frac{\|n(\underline{\mathbf{X}}) - n(\underline{\mathbf{A}}\ \underline{\mathbf{S}})\|^2}{2M} \leq 1$. Note, that this factorisation term is equally suitable to measure the quality of the factorisation $\underline{\mathbf{X}} = \underline{\mathbf{A}}\ \underline{\mathbf{S}}$ as the measure (7.3).

For the sparse NMF problem at hand, the sparseness σ of a row $\mathbf{s}_n$ of $\underline{\mathbf{S}}$ is defined as the fraction of its zero to its nonzero elements. However, as in real life experiments measurements are always corrupted by noise, also

small nonzero entries of a vector should be treated as zero elements. Hence, a nonnegative threshold τ is used which defines the maximum value an entry of $\mathbf{s}_n$ may have in order to be regarded as a zero element. This leads to the following sparseness measure σ:

$$\sigma(\mathbf{s}_n) = \frac{\text{number of elements of } \mathbf{s}_n < \tau}{\text{number of elements of } \mathbf{s}_n}, \tag{7.6}$$

The range of this sparseness measure is $[0, 1]$ whereas $\sigma(\mathbf{s}_n) = 0$ means that $\mathbf{s}$ contains no nil entry while $\sigma(\mathbf{s}) = 1$ indicates that all entries of $\mathbf{s}$ vanish. Hence, $\sum_{n=1}^{N} \sigma(\mathbf{s}_n)$ may maximally become N such that the range of the sparseness term in (7.4) is $[0, 1]$.

Finally, λ is a weighting factor balancing the factorisation and the sparseness term in (7.4). Such weighting factors are usually critical as, for instance, too small values for λ lead to standard NMF while for too large values the sparseness term dominates such that $\underline{\mathbf{X}}$ may only be poorly factorised by $\underline{\mathbf{A}}$ and $\underline{\mathbf{S}}$. However, once an appropriate λ has been determined it should be suitable for a large range of sNMF problems as both the factorisation as well as the sparseness term are normalised. Hence, no new calibration of λ should be necessary when the sizes of the matrices $\underline{\mathbf{X}}$ or $\underline{\mathbf{S}}$ change.

7.2.2 sNMF Algorithm

As the target function defined in Eq. 7.4 is noncontinuous and has many local minima (see Fig. 7.6) we use a Genetic Algorithm (GA) for its minimisation.

GAs are stochastic global search and optimisation methods inspired by natural biological evolution. The core of a GA is a population of possible

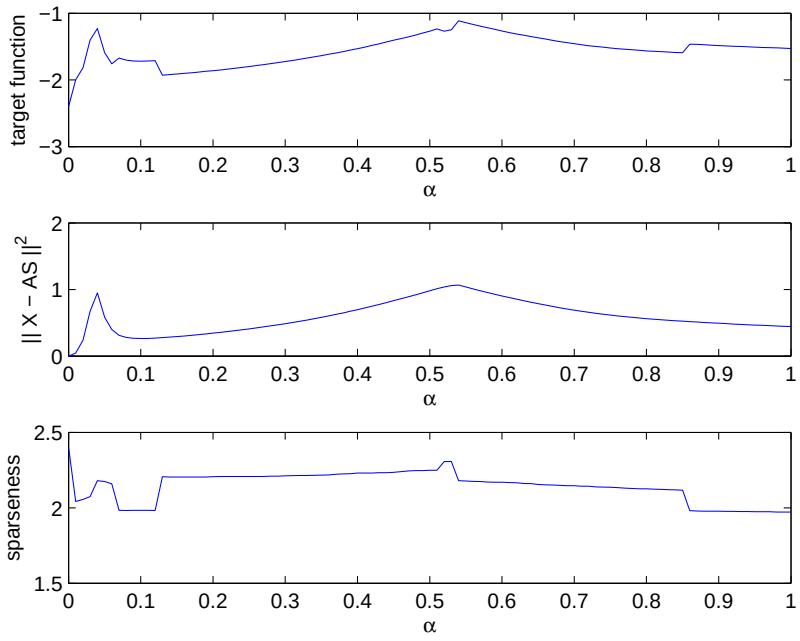

Figure 7.6: Typical plot of the search space associated with (7.4). For this plot a 3×1000 nonnegative random matrix $\underline{\mathbf{S}}$ was generated whose columns had sparsenesses σ of 0.9, 0.8 and 0.7, respectively. This matrix was multiplied by a random nonnegative 3×3 matrix $\underline{\mathbf{A}}$ in order to obtain the matrix of observations $\underline{\mathbf{X}}$. Furthermore, a random nonnegative matrix $\underline{\mathbf{A}}^{rand}$ was generated and the difference $\Delta\underline{\mathbf{A}} = \underline{\mathbf{A}} - \underline{\mathbf{A}}^{rand}$ between $\underline{\mathbf{A}}$ and $\underline{\mathbf{A}}^{rand}$ was determined. Next, for $\alpha = [0, 0.01, 0.02, \ldots, 0.99, 1]$ the matrices $\underline{\mathbf{A}}(\alpha) = \underline{\mathbf{A}} - \alpha\Delta\underline{\mathbf{A}}$ were computed. The inverses of these matrices were multiplied with $\underline{\mathbf{X}}$ to obtain the matrices $\underline{\mathbf{S}}(\alpha)$, i.e. $\underline{\mathbf{S}}(\alpha) = \underline{\mathbf{A}}(\alpha)^{-1}\underline{\mathbf{X}}$. Any negative entries of the matrices $\underline{\mathbf{S}}(\alpha)$ were set to zero in order to fulfil the nonnegativity constraints . For each α the matrices $\underline{\mathbf{A}}(\alpha)$ and $\underline{\mathbf{S}}(\alpha)$ were inserted into (7.4). In the top of this figure the function values of $E(\underline{\mathbf{A}}(\alpha), \underline{\mathbf{S}}(\alpha))$ for the different α's are shown. In the middle, only the reconstruction $\|\underline{\mathbf{X}} - \underline{\mathbf{A}}(\alpha)\underline{\mathbf{S}}(\alpha)\|^2$ is depicted while in the bottom the sparseness term of $E(\underline{\mathbf{A}}(\alpha), \underline{\mathbf{S}}(\alpha))$ is presented (here $\lambda = 1$ was used). Note that for $\alpha = 0$, $\underline{\mathbf{A}}(\alpha) = \underline{\mathbf{A}}$ and $\underline{\mathbf{S}}(\alpha) = \underline{\mathbf{S}}$. Hence, the reconstruction error is nil, the sparseness reaches its maximum and $E(\underline{\mathbf{A}}, \underline{\mathbf{S}})$ is minimal. For $\alpha = 1$ $\underline{\mathbf{A}}(\alpha)$ equals $\underline{\mathbf{A}}^{rand}$ such that the function values shown above can be seen as "walk" on the searchspace from $\underline{\mathbf{A}}^{rand}$ to $\underline{\mathbf{A}}$. As can be seen there are several local minima and the global minima is rather steep and hence hard to find by optimisation algorithms.

solutions, called individuals, to a given optimisation problem as well as a set of operators borrowed from natural genetics. At each generation of a GA, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and reproducing them using the genetically motivated operators. This process leads to the evolution of populations of individuals that better solve the optimisation problem than the individuals from which they were created. Finally, this process should lead to the optimal solution of the optimisation problem even if many suboptimal solutions exist, i.e. if the target function to be optimised has many local minima.

For the minimisation of the target function in Eq. 7.4 the m^2 elements of the solution matrix $\hat{\mathbf{A}}$ have to be determined. Taking advantage of the scaling indeterminacy inherent in the linear mixture model we may assume that the columns of the original mixing matrix $\mathbf{A}$ are normalised such that its diagonal elements are ones. Hence, only the $m^2 - m$ off elements of the matrix $\hat{\mathbf{A}}$ have to be determined by the GA. Accordingly, each of the N_{ind} individuals of the GA algorithm consists of $m^2 - m$ parameters which are usually referred to as genes. As the original mixing matrix is known to have only nonnegative entries it seems self-evident to confine the genes to be nonnegative, too. However, we allow the genes to be negative throughout the optimisation procedure as we have observed in our experiments that otherwise the GA often fails to find the global minimum of the target function.

In every generation of the GA, the fitness of each individual for the optimisation task has to be computed in order to determine the number of offsprings it will be allowed to produce. For this purpose, the target func-

tion (7.4) is evaluated for all individuals. These function values are not used directly as fitness values as otherwise the fittest individuals often produce too many offsprings such that the needed diversity in the population is destroyed and the algorithm converges prematurely to a suboptimal solution. Hence, we use a linear scaling procedure to transform target function values to fitness values.

In order to compute the target function values, for every individual a matrix $\tilde{\mathbf{A}}_-$ is generated whose off elements consist of the genes as stored in the individual and which diagonal elements are set to one. As in the next step the matrix $\tilde{\mathbf{A}}_-$ has to be inverted, care must be taken that it is not singular. Therefore, matrices $\tilde{\mathbf{A}}_-$ which a singular to machine precision are replaced by a nonsingular random matrix with ones on its diagonal. Accordingly, the genes of the corresponding individual are adjusted.

Next, the matrices $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{S}}$ are needed in order to evaluate the target function (7.4). For this purpose, the inverse $\tilde{\mathbf{W}}_-$ of $\tilde{\mathbf{A}}_-$ is computed and the matrices $\tilde{\mathbf{S}}$ and $\tilde{\mathbf{A}}$ are then obtained by setting the negative elements of the matrices $\tilde{\mathbf{S}}_- = \tilde{\mathbf{W}}_-\mathbf{X}$ and $\tilde{\mathbf{A}}_-$, respectively, to zero.

After inserting the matrices $\tilde{\mathbf{S}}$ and $\tilde{\mathbf{A}}$ into (7.4) the resulting target function value is assigned to the corresponding individual. The individuals are then arranged in ascending order according to their target function values and their fitness values $F(p^{(i)})$, $i = 1, \ldots, N_{ind}$, are determined by

$$F(p^{(i)}) = 2 - \mu + 2(\mu - 1)\frac{p^{(i)} - 1}{N_{ind} - 1}, \tag{7.7}$$

where $p^{(i)}$ is the position of individual i in the ordered population. The scalar

parameter μ, which is usually chosen to be between 1.1 and 2.0, denotes the selective pressure towards the fittest individuals.

We have used Stochastic Universal Sampling (SUS) to determine the absolute number of offsprings an individual may produce. Thereby, an arc R_i of length $F(p^{(i)})$ is assigned to the i-th individual, $i = 1, \ldots, N_{ind}$, on a circle of circumference $C = \sum_{i=1}^{N_{ind}} F(x^{(i)})$. Starting from a randomly selected position, $2N_{off}$ marker points are allocated on the circle, whereas the distance between two consecutive marker points is $C/2N_{off}$ and N_{off} is the total number of offsprings to be created. The i-th individual may then produce as many offsprings as there are marker points in its corresponding arc R_i on the circle.

The offsprings are created in a two step procedure. In the first step, two individuals, which are eligible for reproduction according to the SUS criterion, are chosen at random and are used to create a new individual. Thereby, the genes of the new individual are generated by uniform crossover, i.e. each gene of the new individual is created by copying, each time with a probability of 50 %, the corresponding gene of the first or the second parent individuum.

In the second step, called mutation, the actual offsprings are obtained by altering a certain fraction r_{mut} of the genes of the new individuals. These genes are chosen at random and are increased or decreased by a random number in the range of $[0, m_{max}]$. The role of mutation is often seen as providing a guarantee that the probability of searching any given parameter set will never be zero and acting as a safety net to recover good genetic material that may be lost through the action of selection and crossover.

The last action occurring during each generation of a GA is the replacement of the parent individuals by their offsprings. We use an elitist reinsertion scheme meaning that a certain fraction r_{elit} of the fittest individuals is deterministically allowed to propagate through successive generations. Hence, only the $(1 - r_{elit})N_{ind}$ less fittest parent individuals are replaced by their fittest offsprings which ensures that the best solution found so far remains in the population.

In order to keep the algorithm from converging permaturely we make use of the concept of multiple populations. Thereby, a number N_{pop} of populations, each consisting of N_{ind} individuals, are propagating independently in parallel and are only allowed to exchange their fittest individuals after every T_{ex}-th generation. Hence, as long as not all populations have converged to the same solution they will regain some diversity after every T_{ex}-th iteration step. We use the complete net structure scheme for the exchange of individuals which means that every population is exchanging a fraction r_{mig} of its fittest individuals with all other populations.

7.2.3 Parallelisation of sNMF

In the GA used in sNMF the population is divided into several subpopulations which evolve independently for a fixed period of generations. Hence, the evolution of these subpopulations can be carried out by several computers in parallel which only have to communicate during the migration phase of the GA. Such a parallelisation reduces the overall computation time of GA drastically and has therefor also been used in the implementation of sNMF.

Thereby, one of the computers in the cluster is assumed to be the master which is responsible for the distribution and collecting data. In the beginning of the algorithm this master is responsible for loading the data matrix $\underline{\mathbf{X}}$ and for initialising the overall population of the GA. This population is then divided into several subpopulations which are sent to the individual computers together with the observation matrix $\underline{\mathbf{X}}$.

Once arrived there, the individual computers compute the evolution of the individual subpopulations for a certain number of generations until the migration phase of the algorithm starts. In this phase, the master computer collects the individual subpopulations as well as their individual fitness values and carries out the migration steps, i.e. it exchanges some of the fittest individuals of the particular subpopulations. After this migration step is finished the master distributes the new subpopulations to the again to the individual computers and the procedure is repeated. After the algorithm has converged, the master computer finally collects all subpopulations and their fitness values and returns the fittest individual to the user.

In the case of the sNMF, which was written entirely in C, the C implementation of the Message Passing Interface (MPI) standard, called MPICH, was used for the communication between individual computers. This implementation allows easy data transfer between individual computers. Being based on SSH, though, the communication between individual computers can be slow. However, the amounts of data being exchanged in sNMF are rather low such that the time needed for transferring data is all but a critical.

7.2.4 Algorithm Repetitions

Despite the use of the mutation operator and multiple populations, the algorithm failed in many experiments to recover the source and mixing matrix after its first run. In order to keep the computational cost of the algorithm reasonable, this problem could not be overcome by simply increasing the number N_{ind} of individuals and N_{pop} of populations to arbitrarily large values. We still managed to achieve satisfying results by applying the algorithm repeatedly. As usually, the algorithm is provided with the observation matrix $\mathbf{X}$ in its first run which is then decomposed into a first estimate of the source matrix $\tilde{\mathbf{S}}^{(1)}$ and the first estimate of the mixing matrix $\tilde{\mathbf{A}}^{(1)}$, i.e. $\mathbf{X} = \tilde{\mathbf{A}}^{(1)}\tilde{\mathbf{S}}^{(1)}$. In order to make use of the suboptimal results already achieved in the first run, the matrix $\tilde{\mathbf{S}}^{(1)}$ is provided to the algorithm instead of the matrix $\mathbf{X}$ in the second run. The matrix $\tilde{\mathbf{S}}^{(1)}$ is then factorised into the matrices $\tilde{\mathbf{A}}^{(2)}$ and $\tilde{\mathbf{S}}^{(2)}$, which means that the matrix $\mathbf{X}$ can now be factorised as $\mathbf{X} = \tilde{\mathbf{A}}^{(1)}\tilde{\mathbf{A}}^{(2)}\tilde{\mathbf{S}}^{(2)}$. This procedure is repeated K times until the newly determined mixing matrix $\mathbf{A}^{(K)}$ differs only marginally from the identity matrix. With this procedure the final estimates of the mixing matrix $\hat{\mathbf{A}}$ and of the source matrix $\hat{\mathbf{S}}$ are determined as

$$\hat{\mathbf{A}} = \prod_{j=1}^{K} \tilde{\mathbf{A}}^{(j)} \tag{7.8}$$

and

$$\hat{\mathbf{S}} = \tilde{\mathbf{S}}^{(K)}, \tag{7.9}$$

respectively, as the matrix $\mathbf{X}$ can be factorised as $\mathbf{X} = \prod_{j=1}^{K} \tilde{\mathbf{A}}^{(j)}\tilde{\mathbf{S}}^{(K)}$.

	$sp(\mathbf{s})$	$\sigma(\mathbf{s})$
$\mathbf{s}_1$	0.76	0.90
$\mathbf{s}_2$	**0.62**	**0.80**
$\mathbf{s}_1^{pseu}$	0.76	0.90
$\mathbf{s}_2^{pseu}$	**0.69**	**0.72**

Table 7.1: The sparsenesses sp as defined in Eq. 7.10 and σ as defined in Eq. 7.6 ($\tau = 0$) of the original and the pseudo sources. Note, that contradicting the fact that the number of zero elements of the pseudo source $\mathbf{s}_2^{pseu}$ is lower than that of the original source $\mathbf{s}_2$, the sparseness measure sp reaches a higher value for the second pseudo source than for the second original source.

7.2.4.1 Choice of Sparseness Measure

We want to point out that the often used sparseness measure $sp(\mathbf{s})$ of a T-dimensional vector $\mathbf{s}$

$$sp(\mathbf{s}) = \frac{\sqrt{n} - \sum_{i=1}^{T} |s_i| / \sqrt{\sum_{i=1}^{T} s_i^2}}{\sqrt{n} - 1}, \tag{7.10}$$

where s_i is the i-th component of $\mathbf{s}$, cannot be used for the BSS task even if it measures reasonably the sparseness of vectors with only one nonzero entry ($sp(\mathbf{s}) = 1$) and the sparseness of vectors where all elements are equal ($sp(\mathbf{s}) = 0$).

To give an example for the ineligibility of the sparseness measure sp, we have generated two nonnegative random sources $\mathbf{s}_1$ and $\mathbf{s}_2$, each consisting of 1000 data points, and have randomly set 90% of the elements of the first source and 80% of the elements of the second source to zero. These two sources were normalised and then used to constitute the rows of the source matrix $\mathbf{S}$ (cf. Fig. 7.7). The matrix of observations $\mathbf{X}$ was obtained by

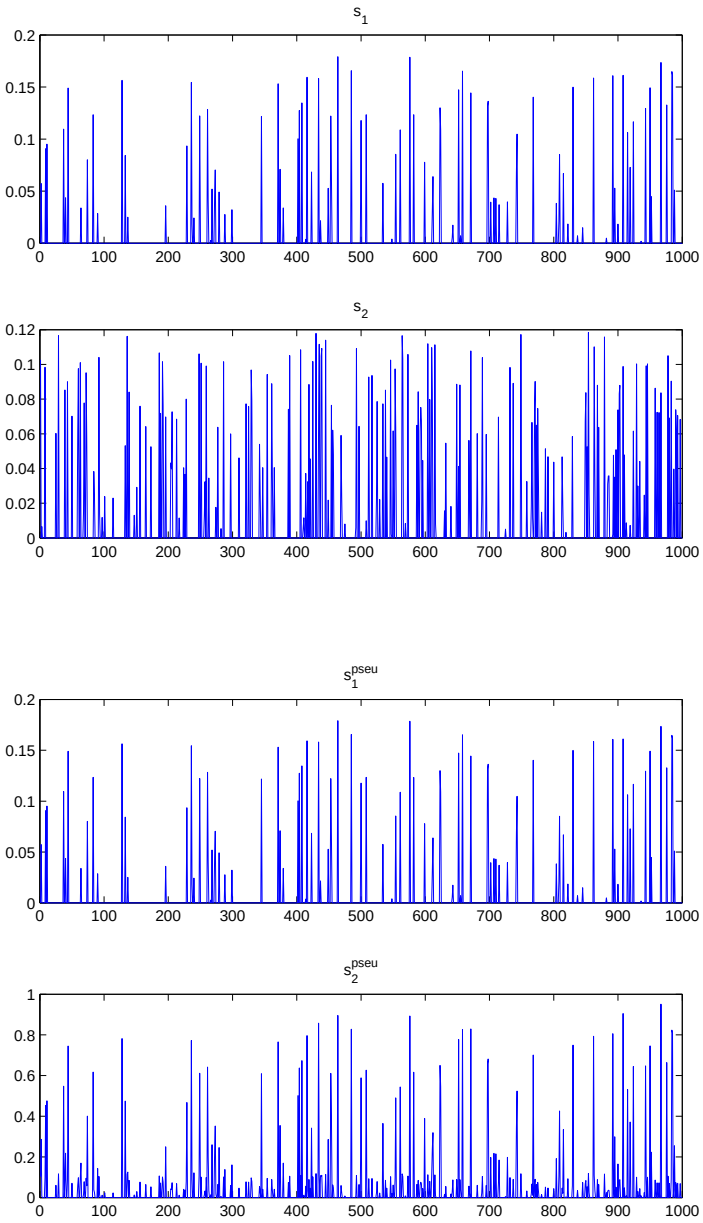

Figure 7.7: Top: The original sources $\mathbf{s}_1$ and $\mathbf{s}_2$. Bottom: The pseudo sources $\mathbf{s}_1^{pseu}$ and $\mathbf{s}_2^{pseu}$. Even if the number of zero elements in $\mathbf{s}_2^{pseu}$ is lower than in the original source $\mathbf{s}$ the sparseness measure sp assigns to it a higher value than to the original source.

mixing the sources with the following mixing matrix

$$\mathbf{A} = \begin{bmatrix} 5 & 1 \\ 6 & 1 \end{bmatrix} \tag{7.11}$$

according to Eq. **??**.

Note, that as the first source is dominating in both of the mixtures the following alternative factorisation of the observation matrix $\mathbf{X}$ is feasible. First, the original mixing matrix $\mathbf{A}$ can be replaced by the pseudo mixing matrix

$$\mathbf{A}^{pseu} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}. \tag{7.12}$$

Correspondingly, the original source matrix $\mathbf{S}$ has then to be replaced by the matrix $\mathbf{S}^{pseu}$, which rows are constituted by the following pseudo sources (see Fig. 7.7)

$$\mathbf{s}_1^{pseu} = \mathbf{s}_1, \tag{7.13}$$

$$\mathbf{s}_2^{pseu} = 5\mathbf{s}_1 + \mathbf{s}_2. \tag{7.14}$$

Obviously, these matrices also factorise $\mathbf{X}$, i.e. $\mathbf{X} = \mathbf{A}^{pseu}\mathbf{S}^{pseu}$ still holds, but the number of zero elements in the second pseudo source $\mathbf{s}_2^{pseu}$ is about 8% lower than that of the original source $\mathbf{s}_2$ (cf. Tab. 7.1). Hence, a BSS algorithm based on the sparseness measure σ as defined in Eq. 7.6 would correctly favor the original source and mixing matrix $\mathbf{S}$ and $\mathbf{A}$, respectively, over their pseudo variants $\mathbf{S}^{pseudo}$ and $\mathbf{A}^{pseudo}$.

In contrast, the sparseness measure *sp* as defined in Eq. 7.10 assigns

	c_1	c_2	c_3	CTE
sparse NN BSS	1.00	1.00	1.00	0.39
fastICA	1.00	1.00	0.75	5.19

Table 7.2: Results obtained with the presented method (abbreviated as sparse NN BSS) and the fastICA algorithm. Displayed are the correlation coefficients c_i between the i-th original source and its corresponding estimate as well as the cross-talking error (CTE) between the estimated and the original mixing matrix.

a higher sparseness value to the second pseudo source $\mathbf{s}_2^{pseudo}$ than to the original source $\mathbf{s}_2$ (cf. Tab. 7.1). Accordingly, a sparse BSS algorithm based on this sparseness measure would fail to recover the original source and mixing matrix $\mathbf{A}$ and $\mathbf{S}$, respectively.

7.2.4.2 Recovery of correlated sources

In this section we show that the presented method is capable of solving the BSS problem even if the underlying sources are correlated. This case is especially interesting as a very popular alternative BSS technique, called Independent Component Analysis (ICA), fails in such a situation as it is based on the assumption that the underlying sources are statistically independent.

For the simulation, we have generated three sources $\mathbf{s}_i$, $i = 1, \ldots, 3$, as follows. The first and the second source were generated as nonnegative random vectors where 90% and 80%, respectively, of the elements were randomly set to zero. The third source was generated from the second source by adding a linear function, i.e.

$$\mathbf{s}_3(n) = \mathbf{s}_2(n) + 0.001(n-1), \tag{7.15}$$

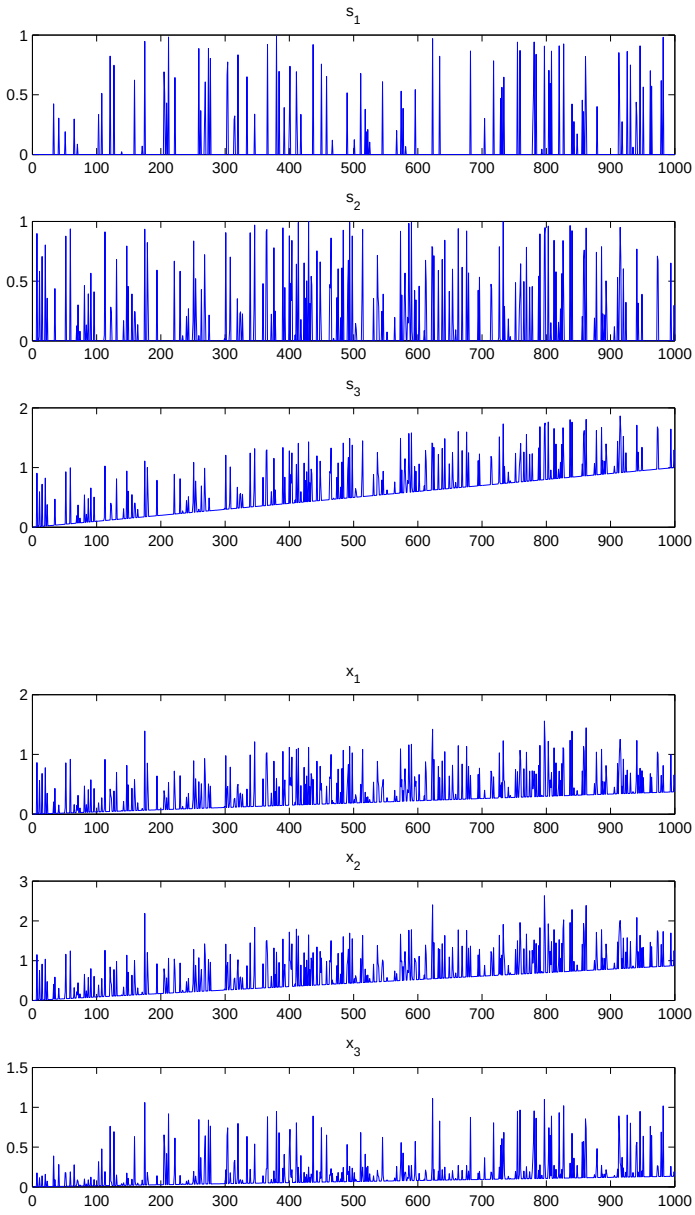

Figure 7.8: Top: The original sources $\mathbf{s}_i$. Note, that $\mathbf{s}_3$ was obtained from $\mathbf{s}_2$ by adding a linear function. Bottom: The rows $\mathbf{x}_i$ of the mixture matrix $\mathbf{X}$ as provided to the algorithms.

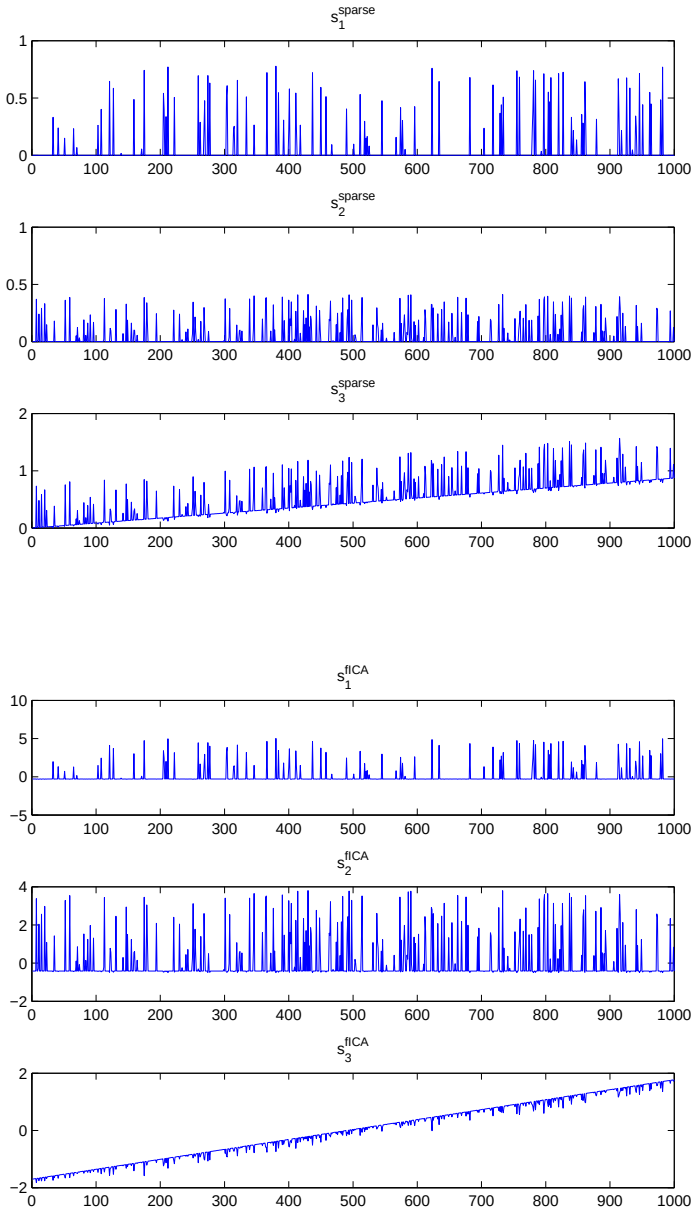

Figure 7.9: Top: The estimates $\mathbf{s}_i^{sparse}$ of the sources as obtained by the nonnegative sparse BSS algorithm. Bottom: The estimates $\mathbf{s}_i^{fICA}$ of the sources as obtained by the fastICA algorithm. Note, that fastICA fails to recover the third source.

where $\mathbf{s}_i(n)$ denotes the n-th element of the source $\mathbf{s}_i$ and $n = 1, \ldots, 1000$ (cf. Fig. 7.8).

This procedure lead to a non-vanishing correlation coefficient of $c = 0.65$ between the second and the third source, while the sources $\mathbf{s}_1$ and $\mathbf{s}_2$ as well as $\mathbf{s}_1$ and $\mathbf{s}_3$ were uncorrelated. As before, these sources were used to constitute the source matrix $\mathbf{S}$.

The observation matrix $\mathbf{X}$ (cf. Fig. 7.8) was generated as in Eq. **??** by multiplying the source matrix $\mathbf{S}$ with the following well conditioned (condition number about 5.5) mixing matrix

$$\mathbf{A} = \begin{bmatrix} 0.4554 & 0.5833 & 0.3739 \\ 0.8916 & 0.3988 & 0.8736 \\ 0.9042 & 0.0604 & 0.1326 \end{bmatrix}. \tag{7.16}$$

When we used the presented nonnegative sparse BSS algorithm with the parameters set as in the last section, we could recover the sources as well as the mixing matrix almost perfectly as can be seen in Tab. 7.2 and Fig. 7.9. Thereby, only $K = 2$ successive runs of the algorithm were needed.

In contrast, such a perfect recovery seems to be impossible by ICA based BSS. To show this, we have used the famous fastICA algorithm in order to recover the sources $\mathbf{s}_i$, $i = 1, \ldots, 3$, and the mixing matrix $\mathbf{A}$. This algorithm also succeeded in recovering the sources $\mathbf{s}_1$ and $\mathbf{s}_2$ almost perfectly, but it failed to recover the third source $\mathbf{s}_3$ (cf. Fig. 7.9). Accordingly, the cross-talking error between the estimated and the original mixing matrix is more than five times higher than the CTE achieved with the sparse nonnegative

BSS approach (cf. Tab. 7.2). Surely, these poor results are not surprising as we have violated the independence assumption by creating correlated sources.

Hence, the presented algorithm seems to be capable of solving BSS problems where other well established BSS algorithms, like fastICA, principally fail.

7.2.5 Reliability of sNMF

As the search space associated with the target function (7.4) is rather complex (cf. Fig. 7.6) the parameters of the GA used in sNMF have to chosen deliberately in order to obtain high reliability. For this purpose, various simulations have been carried out in order to determine such a set of parameters. Furthermore, a suitable value for the weighting factor λ in (7.4) was determined.

7.2.5.1 sNMF applied to 3×3 BSS problems

In the first set of simulations, the sNMF algorithm was used to solve a 3×3 BSS problem. Thereby, 25 different 3×1000 nonnegative random source matrices $\underline{\mathbf{S}}_k$, $k = 1, \ldots, 25$, were generated, whereas the sparsenesses of the individual rows of these matrices were chosen at random from the interval $[0.7 \ldots 0.9]$. These sparseness values may seem large at first glance but are to be expected in real life microarray experiments.

Furthermore, 25 nonnegative random 3×3 mixing matrices $\underline{\mathbf{A}}_k$ were generated such that 25 artificial observation matrices $\underline{\mathbf{X}}_k$ could be computed as $\underline{\mathbf{X}}_k = \underline{\mathbf{A}}_k \underline{\mathbf{S}}_k$.

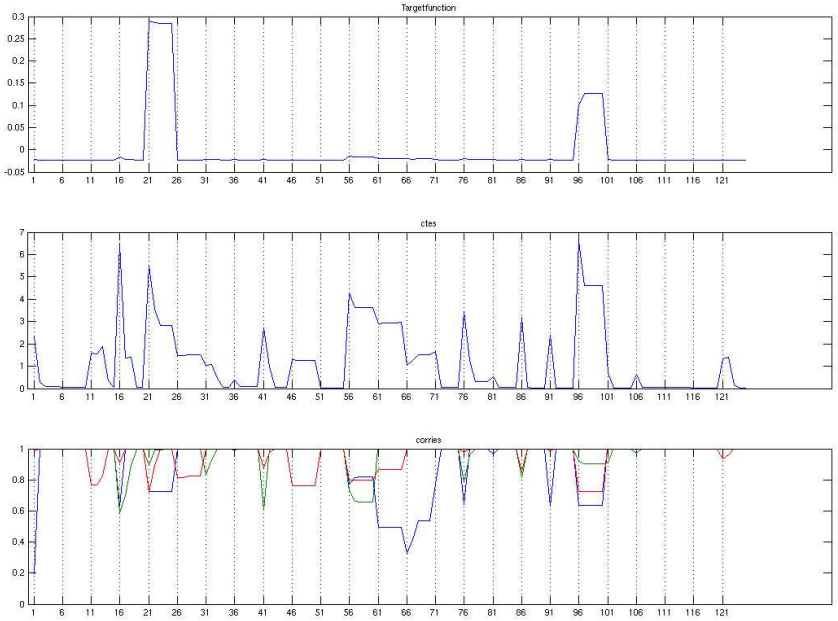

Figure 7.10: Results as obtained by sNMF when applied to 25 toy data sets. Top: Values of the target function (7.4). Middle: CTEs between original and recovered mixing matrix. Bottom: CC between original and recovered sources. Vertical lines separate the results from different data sets. For each data sets five data points have been recorded according to the five repetitions of the algorithm.

parameter	value
λ (in (7.4))	0.01
τ (in (7.6))	0.01
nr. of subpopulations	8
subpopulation size	50
migration rate	0.2
generations between migrations	100
nr. of migrations	20
crossover rate	1
mutation rate	0.0001
generation gap	1
Rank min	0.5
repetitions of algorithm per dataset	5

Table 7.3: Parameters used by sNMF when applied to 3×3 problems

source nr.	1	2	3	4	5
sparseness σ	0.7790	0.7030	0.7140	0.9070	0.7940

Table 7.4: Sparsenesses of the individual sources used in the 5×5 experiment.

These matrices were provided to the sNMF algorithm such that 25 estimates $\underline{\tilde{\mathbf{A}}}_k$ and $\underline{\tilde{\mathbf{S}}}_k$ of the source and the mixing matrices, respectively, were be obtained. In order to quantify the obtained results the Amari cross-talking error (CTE) between the original and the recovered mixing matrix was computed. Furthermore, the correlation coefficient (CC) between the recovered and the original sources (i.e. the rows of $\underline{\mathbf{S}}_k$ and $\underline{\tilde{\mathbf{S}}}_k$) was determined. Thereby, a recovered source matrix $\underline{\tilde{\mathbf{S}}}_k$ was considered to be a valid estimate of the original matrix $\underline{\mathbf{S}}_k$ if the smallest CC between the original and the recovered sources was larger than 0.95. Likewise, the recovery of the mixing matrix was considered successful, if the CTE between the original and the recovered matrix was smaller than 1.

The best results were obtained with the parameters as listed in Table 7.3. Thereby, 18 of the 25 mixing matrices $\underline{\mathbf{A}}_k$ and source matrices $\underline{\mathbf{S}}_k$ could be recovered successfully as can be seen in Fig. 7.10

7.2.5.2 sNMF applied to 5×5 datasets

In order to investigate how the parameters of the GA as well as the weighting factor λ (in (7.4)) have to be changed when the dimension of the BSS problem increases, sNMF was also applied to a 5×5 BSS problem. Thereby, the individual sources to be recovered had sparsenesses σ as depicted in Tab. 7.4 and consisted of 1000 data points (see Fig. 7.11 left). These sources were

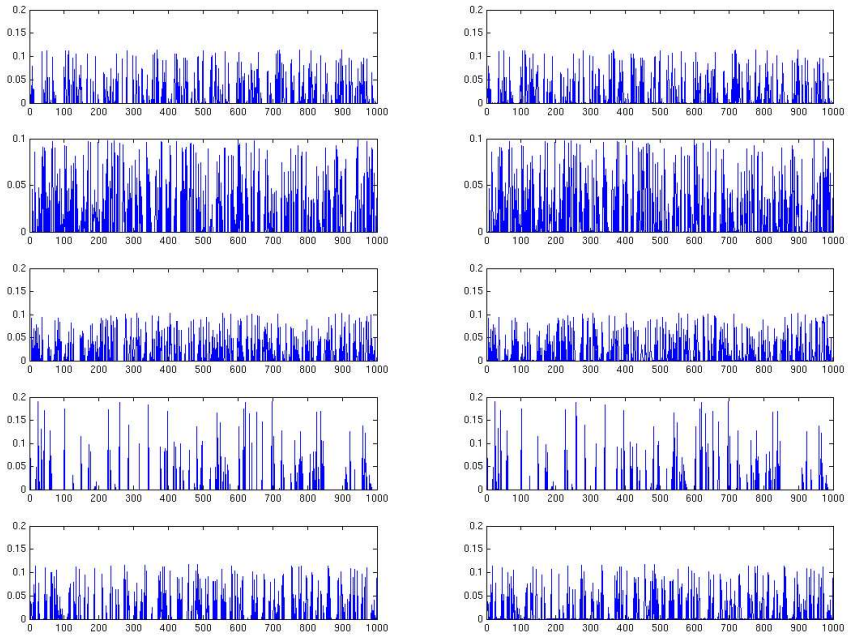

Figure 7.11: The original sources (left) and their estimates as obtained by sNMF (right). In order to be easy to compare, both the original and the recovered sources were normalised.

mixed by the following 5×5 nonnegative mixing matrix

$$\underline{\mathbf{A}} = \begin{bmatrix} 0.451847 & 0.092713 & 0.988865 & 0.821230 & 0.343631 \\ 0.392468 & 0.081602 & 0.554274 & 0.085562 & 0.923546 \\ 0.910548 & 0.105504 & 0.531529 & 0.501901 & 0.879989 \\ 0.565076 & 0.593526 & 0.616349 & 0.676884 & 0.340933 \\ 0.822875 & 0.912405 & 0.859161 & 0.420878 & 0.797631 \end{bmatrix} \tag{7.17}$$

to obtain the matrix of observations $\underline{\mathbf{X}}$ (see Fig. 7.12). This observation matrix was then provided to the sNMF algorithm whereas the parameters

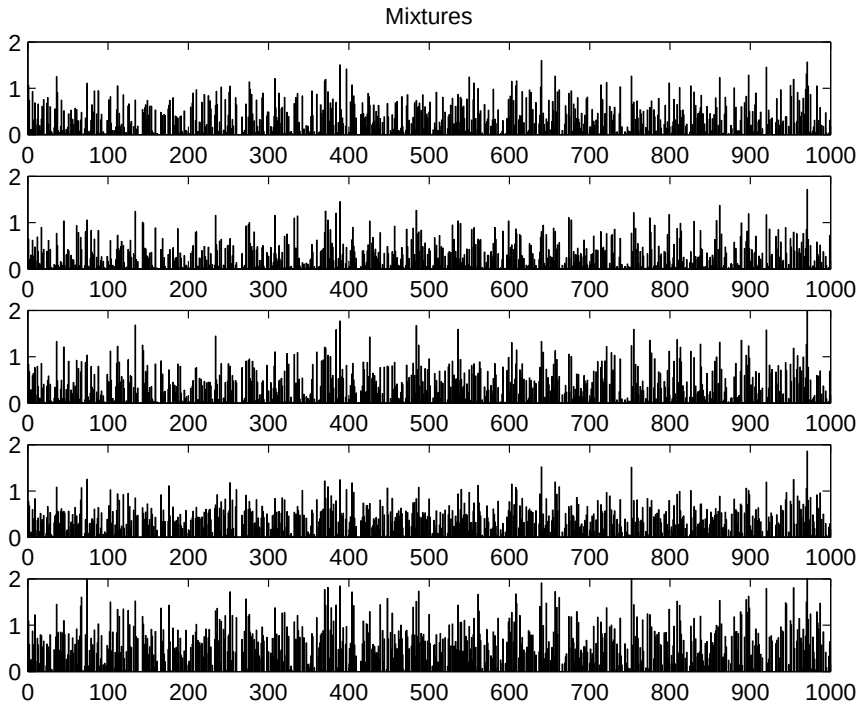

Figure 7.12: The rows of the observation matrix $\underline{\mathbf{X}}$ as provided to the sNMF algorithm.

as listed in Tab. 7.5 were used. As can be seen in the right of Fig. 7.11 the sources were recovered perfectly. Accordingly, the correlation coefficient between the original and the recovered sources were always larger than 0.99.

Furthermore, the CTE between the original and the estimated mixing matrix was only 0.1825 thus indicating an excellent recovery.

Comparing the parameters used in the 3×3 BSS problem (see last section) with the parameters used here three differences appear: first, the number of individuals per subpopulation is two times higher than in the case of the 3×3 problem. Furthermore, 3.75 times more subpopulations are used. This increase is necessary as also the number of parameters to be determined by

parameter	value
λ (in (7.4))	0.001
τ (in (7.6))	0.01
nr. of subpopulations	30
subpopulation size	100
migration rate	0.2
generations between migrations	100
nr. of migrations	20
crossover rate	1
mutation rate	0.0001
generation gap	1
Rank min	0.5
repetitions of algorithm per dataset	5

Table 7.5: Parameters used by sNMF when applied to 5×5 problem.

sNMF has increased considerably (6 in the 3×3 case compared with 20 in the 5×5 case).

Furthermore, although the reconstruction and the sparseness term in (7.4) are normalised, the weighting factor λ had to be set to a ten times smaller value compared with that used in the 3×3 problem. When larger values were used the reconstruction error increased such that no satisfactory factorisation of $\underline{\mathbf{X}}$ was be obtained.

7.2.6 Application of sNMF to Real World Data

We have applied the proposed sparse NMF algorithm to analyse microarray data which were recorded during an investigation of pseudoxanthoma elasticum (PXE), an inherited connective tissue disorder characterised by progressive calcification and fragmentation of elastic fibers in the skin, the retina, and the cardiovascular system. During the investigations M=8 microarray experiments have been carried out. In the first and the second experiment the PXE fibroblasts were incubated in Bovine Serum Albumin (BSA) whereas the incubation time was three hours in the first and 24 hours in the second experiment. In the third experiment the PXE fibroblasts were incubated for three hours in an environment with a high concentration of the Transcription Growth Factor beta and in the fourth experiment the cells were incubated for 24 hours in an environment which was rich in Interleukin 1 beta. The same experiments were then repeated with a control group of normal fibroblasts. The used Affymetrix HG-U133 plus 2.0 microarray chips are capable of detecting the expression levels of more than 54675 genes, of which, however, only T=10530 were expressed significantly in at least one of the experiments. Hence, only these 10530 genes were considered in the further data analysis.

This data set was used to constitute the 8×10530 observation matrix $\mathbf{X}$ which was then decomposed into the matrices $\hat{\mathbf{A}}$ and $\hat{\mathbf{S}}$ by the proposed sparse NMF algorithm. For the genetic algorithm we increased the number of sub-populations to $N_{pop} = 56$, the maximum number of iterations to 2500 and the number of algorithm repetitions to $K = 8$ while the remaining

Source	1	2	3	4	5	**6**	7	8
#(cib)	0	13	7	0	4	**35**	9	6

Table 7.6: Number of genes related with calcium ion binding (#(cib)) for each of the eight estimated sources. Only genes which are rated exclusively as having a calcium ion binding molecular function in the Gene Ontology [26] database were considered. Most genes related with calcium ion binding are clustered into source 6.

parameters were set as in the last section. Note, that after the fifth of the $K = 8$ repetitions of the algorithm the resulting matrices $\hat{\mathbf{A}}$ and $\hat{\mathbf{S}}$ did not change any further which indicated the overall convergence of the algorithm. After the sparse NMF analysis each row of the matrix S should ideally consist of the genetic fingerprint of one specific biological process only. It must be noted, however, that at least one hundred of such processes are occurring simultaneously in a biological cell while the number of available observations and hence the number of estimated sources was only eight. But despite these highly overcomplete settings the algorithm succeeded in grouping the majority of genes which are related with calcium ion binding (344 in total) and hence with the disease picture of PXE into the sixth estimated source (see Tab. 7.6). Furthermore, the calcium ion binding related genes in the sixth source seem to be specific for only one biological process as maximally 14 % of them could be found in any of the remaining sources.

For comparison, we have used Independent Component Analysis (ICA) to factorise the observation matrix $\mathbf{X}$ into the $m \times m$ mixing matrix $\mathbf{A}_{ICA}$ as well as into $m \times T$ source matrix $\mathbf{S}_{ICA}$. In ICA the nonnegativity and the sparseness constraints are replaced by the assumption that the rows of the matrix $\mathbf{S}_{ICA}$ are mutually independent. We have chosen the well-known fas-

tICA algorithm [44] for the data analysis. In contrast to the results obtained with the sparse NMF algorithm, maximally 12 genes related with calcium ion binding could be found in one source. Hence, the proposed sparse NMF algorithm seems to be better suited for the analysis of PXE cells than the fastICA algorithm.

7.3 Summary

In this chapter two new approaches to solve the BSS problem with nonnegativity constraints have been presented. The first, called extended sparse NMF can be used when estimations or approximations of the sparseness of the sources are available *a priory*. As was shown, incorporating such additional knowledge leads to better and more reliable results when compared to standard NMF.

For the second algorithm, called sparse NMF, *a priori* knowledge about the spraseness of the sources is necessary. sNMF automatically factorizes the observation matrix $\underline{\mathbf{X}}$ into $\underline{\mathbf{A}}$ and $\underline{\mathbf{S}}$ such that the rows of $\underline{\mathbf{S}}$ are as sparse as possible. As was shown in several simulations, sNMF is also capable of solving linear BSS problems uniquely as long as the rows of the matrix $\underline{\mathbf{S}}$ are sparsely represented.

sNMF was also applied to real world microarray data. Thereby, it seemed to be more suitable than the fastICA algorithm. However, further experiments with real world data sets are necessary to confirm this first impression.

Chapter 8

Conclusiones y Perspectivas

Con el desarrollo de esta tesis se ha demostrado que es posible llevar a cabo el procesamiento de un gran número de datos obtenidos de experimentos realizados con microarray, los cuales pueden ser analizados utilizando técnicas de factorización. De este modo, un nuevo algoritmo de BSS, llamado factorización dispersa de la matriz no negativa, ha sido desarrollado y con él se puede particularmente, analizar datos de microarray. Este algoritmo se basa en el modelo linear de la mezcla usado comúnmente en BSS y teniendo encuenta las siguientes suposiciones:

1. $\underline{\mathbf{A}}$ y $\underline{\mathbf{S}}$ son no negativas,

2. las filas de $\underline{\mathbf{S}}$ se representan de manera escasa.

Estas dos suposiciones se pueden resuelver claramente en el contexto de datos microarray, pues las cuálas producen solamente cantidades positivas de mRNA, mientras que los procesos celulares conducen solamente a la expresión de un número pequeño de genes.

Como ha sido demostrado en varias simulaciones realizadas, el sNMF a sido superior al estándar NMF pues ha sido capaz de solucionar el problema de BSS. Además, sNMF puede analizar de forma satisfactoria datos de microarray que algoritmos de ICA que actan asumiendo que las filas de $\underline{\mathbf{S}}$ son independientes. Esto es sobre todo debido al hecho que el uso de ICA en datos microarray conduce a menudo a fuentes que contienen el positivo asícomo elementos negativos simultáneamente. Los últimos son, sin embargo, imposibles de interpretar ya que no existe ninguna expresión negativa de genes. Además, la suposición de la independencia hecha en ICA es difícil de justificar desde un punto de vista biológico, mientras que las suposiciones hechas en sNMF resultan biológicamente razonables.

Por consiguiente, el sNMF parece conducir a resultados más razonables cuando se aplica al sistema de datos de PXE que cuando se aplica fastICA. Sin embargo, estos resultados son algo preliminares y requieren de otras investigaciones biológicas con el fin de poder confirmar los resultados. No obstante, pueden ser vistas como una primera introducción para la utilizaciíon de sNMF en datos de microarray usando situaciones del mundo real.

Sin embargo, el sNMF también tiene una desventaja cuando se compara con los algoritmos ya conocidos de ICA, esto debido a que la base del sNMF es un algoritmo genético, los tiempos del cómputo pueden ser considerablemente más largos que los necesitados por e.g. fastICA. Por esta razón, fue desarrollado un algoritmo genético paralelo, de modo que se desarrollan grandes clusters que permiten aumentar así la velocidad de ejecución de manera perceptible.

Comparando análisis estádar como técnicas jerárquicas o k-means clustering, los métodos de análisis BSS tienen generalmente la ventaja que los genes individuales se pueden asignar a más de un cluster. De este modo, esta resulta ser otra conclusión obtenida de las observaciones realizas de donde se puede decir que los genes individuales participan en más de un proceso celular.

Actualmente, los límites del algoritmo propuesto originan del modelo linear subyacente de la mezcla. Como fue demostrado después de una revisión cuidadosa de la expresión y regulación del gene, el modelo se quebranta si los genes individuales son regulados por varios factores de la transcripción o si los procesos celulares reaccionan a los estímulos externos por medio de splicing alternativo. Por otro lado, el trabajo futuro debe centrarse en desarrollar los modelos no lineales de la mezcla capaces de tener en cuenta los mecanismos reguladores antes mencionados.

Hasta este momento, el sNMF fue aplicado solamente en los datos microarray. Pues las suposiciones hechas en sNMF se han realizado de manera general, sin embargo, debe también ser aplicado a otros sistemas de datos que desarrollen tareas biológicas o de descomposición de imagen en el futuro.

Appendix A

Cells and DNA

In this section some of the fundamentals of biological cells as well as of the storage of genetic information will be reviewed. Thereby, the differences between the two basic types of cells, prokaryotes and eukaryotes, will be explained. Furhtermore, it will be shown how genetic information is stored in DNA.

A.1 Biological cells

In this section the main features of prokaryotic and eukaryotic cells will be reviewed for convenience.

A.1.1 Prokaryotic cells

Prokaryotes are supposed to be the chronologically first form of living organisms and are the simplest known cells capable of reproduction. Literally, their name means "before the nucleus" indicating that they emerged before

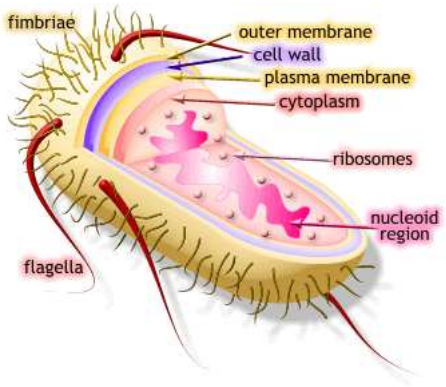

Figure A.1: Prokaryotic cell (image courtesy of www.blob.org)

cells with nuclei evolved. Indeed, prokaryotic cells lack any organelles and are rather simple as can be seen in Fig. A.1

The flagella and fimbriae are used by the prokaryotic cell to move itself as well as any substances around it. Furthermore, the cells use them to attach themselves to surfaces. The outer membrane, the cell wall and the plasma membrane form permeability barriers for the selective passage of nutrients and exclusions of harmful substances. Furthermore, they act as enzymes and mediate specific reactions on the surface of the cell. The cytoplasm, a liquid with high water content, contains granules of e.g. sugar of fat which are important for the energy supply of the cell. It also contains the ribosomes which are responsible for the production of proteins. The nucleoid region contains the hereditary information of the cell in the form of a procaryotic chromosome, i.e. a large circular molecule of DNA (see A.2.1). Note, that the nucleoid region is not separated from the rest of the cell by any membranes, i.e. a nucleus is missing in prokaryotes. Bacteria and Archae are the only examples of prokaryotic cells and are in most cases unicellular. Generally,

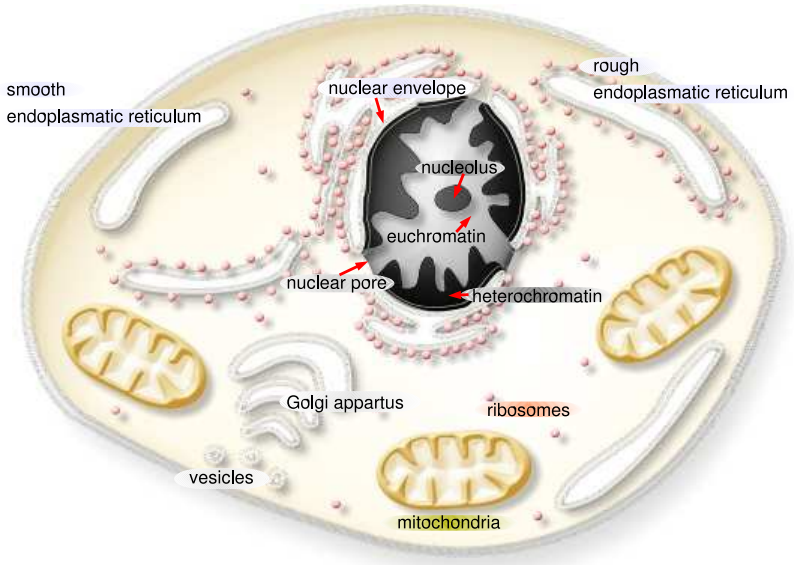

Figure A.2: Eukaryotic cell (image courtesy of www.blob.org)

organisms consisting of prokaryotic cells are called prokaryotes.

A.1.2 Eukaryotic cells

In contrast to prokaryotic cells eukaryotic cells are much complexer and, as can be seen in Fig. A.2, contain a nucleus as well as other organelles. In the nucleus, hereditary information is stored in the form of eu- and heterochromatin (see A.2.1) while ribosomes are produced and matured by one or more of its nucleoli. The nucleus is separated from the cytoplasm by a double membrane called the nuclear envelope. In the nuclear envelope some thousand nuclear pores can be found which selectively allow particles to enter or leave the nucleus. The nuclear envelope is continuous with the rough endoplasmatic reticulum which owes its rough surface to the many ribosomes attached to it. It is responsible for the synthesis of membrane and

organellar proteins as well other as proteins which may leave the cell later. In the Golgi apparatus these proteins are packed into vesicles which finally transport them to the cell membrane. The unbound ribosomes in the cell also produce proteins which are, however, used internally in the cell. Further organelles usually found in eukaryotic cells are the smooth endoplasmatic reticulum and mitochondria. The first is used for the synthesis and digestion of fatty acids and phospholipids while mitochondria convert organic materials into energy (in the form of ATP) and can hence be seen as cellular powerplants. Even not shown in Fig. A.2 eukaryotic cells also have flagella attached to their surface that a variously involved in movement, feeding and sensing. All multicellular organisms like animals, plants and fungi as well as the usually unicellular protists consist of eukaryotic cells. Generally, organisms consisting of eukaryotic cells are called eukaryotes.

A.2 Genetic Coding

In this section, the chemical and spacial structure of deoxyribonucleic acid (DNA) will be explained and it will be shown how DNA molecules are compressed in order to fit into their host cells. Furthermore, it will be explained how genetic information is encoded into DNA.

A.2.1 DNA and RNA

Today it is common knowledge that DNA directs the development of cells and contains hereditary biological information. Basically, DNA is a polynucleotide, i.e. a polymer composed of repeating subunits called nucleotides.

These nucleotides consist of a phosphate group, the 5-carbon sugar 2'-deoxyribose and one of the 4 cyclic nitrogenous bases adenine, thymine, cytosine and guanine (abbreviated A, T, C, G). They are joined by a phosphodiester bond, in which the phosphate, which is attached to the 5' carbon of deoxyribose of the one nucleotide is connected to the hydroxyl group attached to the 3' carbon of the other nucleotide. This creates a chain of alternating phosphate and sugar groups with the basis sticking out the side. Obviously, any chain of nucleotides will have a free 5' phosphate at one end of the chain, and a free 3' hydroxyl at the other end. These are referred to as the 5' and the 3' end, respectively, of the DNA strand. Related with this nomenclature of the ends of the DNA are the terms "upstream" and "downstream". A point on the DNA is said to lie upstream with respect to any reference point, if it is closer to the 3' end than the reference point. If it lies closer to the 5' end the point is said to lie downstream.

DNA is hardly found as a single stranded molecule in nature. Usually, it consists of two polynucleotide chains connected together by hydrogen bondings between the bases of the nucleotides. It must be noted, however, that only the bases adenine and thymine as well as guanine and cytosine will hydrogen bond with each other. Hence, in order for two strands of DNA to base pair, the appropriate bases must be present in corresponding positions on the two strands. Such strands are then said to be complementary. The two strands of DNA run in opposite directions when base paired, i.e. the 5' end of the one strand is next to the 3' end of the other strand. Spatially, two-stranded DNA forms a double helix with ladder step like base bondings between the strands.

In higher eukaryotic organisms like the human being tens to hundreds of millions of base pairs are found in a single DNA molecule. These enormous amounts of base pairs would lead to molecules stretching to several centimetres in length if assembled as pure double helices. Hence, the DNA molecules have to be condensed in order to fit into the nuclei of the cells. In eukaryotic cells this condensation is achieved with the aid of the histone molecules H2A, H2B, H3, H4 and H1. Thereby, two copies of each of the first four histones are used to build a nuclear core protein, called nucleosome core, around which 147 base pairs of DNA are wrapped. These bases together with the nucleosome core are often called a nucleosome. Usually, 20 to 60 base pairs of DNA link one nucleosome with the next whereas each linker region is occupied by a single molecule of the histone H1. Collectively, DNA together with its associated histones is usually called chromatin.

Even after nucleosomes are formed the DNA molecules are still too large to fit into the nuclei of the cells. However, chromatin is folded as its histone molecules interact with each other. Thereby, two different structures of chromatine, called heterochromatin and euchromatin, appear. As will be discussed later, not all parts of the DNA are actually used by cells. These so-called non-coding parts are very densly packed and constitute the heterochromatin. In contrast, euchromatin contains the coding parts of the DNA and is only loosely-packed into loops of 30 nm fibers. The overall level of condensation varies throughout the cell cycle and reaches its maximum during mitoses. At this stage the densly packed chromatin is usually called chromosome and can be made visible with an optical microscope.

In prokaryotic cells the number of base pairs constituting the DNA molecule

is usually much lower than in eukaryotic cells. Still, the DNA is bound to histone like proteins and also appears in a supercoiled structure. In contrast to eukaryotic DNA prokaryotic DNA is circular, i.e. its ends are fixed together.

Ribonucleic acid (RNA) is another important nucleic acid found in biological cells. Its structure is similar to the structure of DNA, however, RNA is mostly found as a single stranded molecule even if it occasionally occurs double-stranded. Furthermore, the sugar 2'-deoxyribose found in DNA is replaced by ribose in RNA. As in DNA the bases adenine, cytosine and guanine can be found in RNA while thymine is replaced by uracil. Like thymine bases these uracil bases pair with adenine. The nucleotides of the RNA are usually referred to as ribonucleotides and are abbreviated as ATP, GTP, UTP and CTP depending on which of the bases are bound to them.

The most common types of RNA found in biological cells are messenger RNA (mRNA), transfer RNA (tRNA) and ribosomal RNA (rRNA). tRNA is used in the synthesis of proteins while rRNA is one of the major components of ribosomes. mRNA, on the other hand, is used as an transmitter of genetic information. All of them will be explained in more detail in later sections.

A.2.2 The Genetic Code

Generally speaking the function and structure of a biological cell is defined by the types of proteins it produces. A skin cell, for instance, will obtain its particular properties by producing other types of proteins than, e.g., a liver cell. The cells generate their specific proteins by producing amino acids which are then linked by covalent peptide bonds. Even proteins are usually

large molecules with molecular weights ranging from about 10000 to 100000 only 20 different kinds of amino acids are used to build them. The blueprints of how these 20 amino acids are assembled to build the protein are stored in the DNA.

In DNA, any information is encoded by the order in which the bases adenine (A), guanine (G), cytosine (C) and thymine (T) appear. Thereby, a group of three of these bases, called a codon, is used to encode exactly one type of amino acid. Obviously, 64 amino acids could be encoded by all possible codons while only 20 amino acids can be found in proteins. Still, all codons code for a specific type of amino acid, which means, in turn, that some of the amino acids are coded for by multiple codons. Hence, the genetic code is called degenerated.

In some cases, the redundant codons are related to each other by their sequence. Leucine, for instance, is specified by the codons CUU, CUA, CUC and CUG, i.e. by four codons which vary only in the third nucleotide position. This third position is known as the "wobble" position of the codon as in many cases the identity of the base at the third position can wobble, and the same amino acid will still be specified. This property allows some protection against mutations, i.e. accidental exchanges of one nucleotide by another. If a mutation occurs at the third position of a codon, it is likely that the amino acid in the encoded protein will still not change.

The information for an entire protein is encoded into the DNA as a set of consecutive codons. Thereby, the first codon, also called start codon is AUG and indicates the beginning of the code. The order in which the following cordons appear in the 5' to 3' direction of the DNA defines the order in

which the corresponding amino acids will be assembled during the synthesis of the protein. The last codon in the sequence, also called the termination codon, is either UAA, UAG or UGA and indicates the end of the code. As will be discussed later, not necessarily all codons between the starting and the termination codon are used for the synthesis of proteins in eukaryotic cells. Furthermore, not all parts of the DNA code for proteins. Some blocks may deal as switches for the synthesis of proteins while others contain the information of RNAs needed for the construction of ribosomes or for the assembly of proteins. Finally, the overwhelming majority of the DNA (about 90%) is assumed to be "junk", i.e. does not code for any information at all.

Appendix B

Polymerase Chain Reaction

In the following the PCR process will be discussed (without loss of generality) considering as example the DNA molecule depicted in the upper left of Fig. B.1. Thereby, the fragment to be multiplied (in the following called target fragment) is highlighted in blue. For the sake of clarity, the strand of the target fragment consisting of the sequence ATCTGG will be called upper target strand while the strand of the target fragment consisting of TAGACC will be referred to as lower target strand in the sequel.

In order to multiply the target fragment mentioned above by PCR two important ingredients are required. On the one hand two primers are needed which consist of the nucleotides complementary to those found at the 3' or 5' end, respectively, of the upper target strand. On the other hand, a special DNA polymerase enzyme is needed that is stable even at temperatures high enough to denature DNA. This enzyme, called Taq polymerase, was first isolated from the genetically engineered bacterium *Thermus aquaticus (Taq)* which is found in thermal springs.

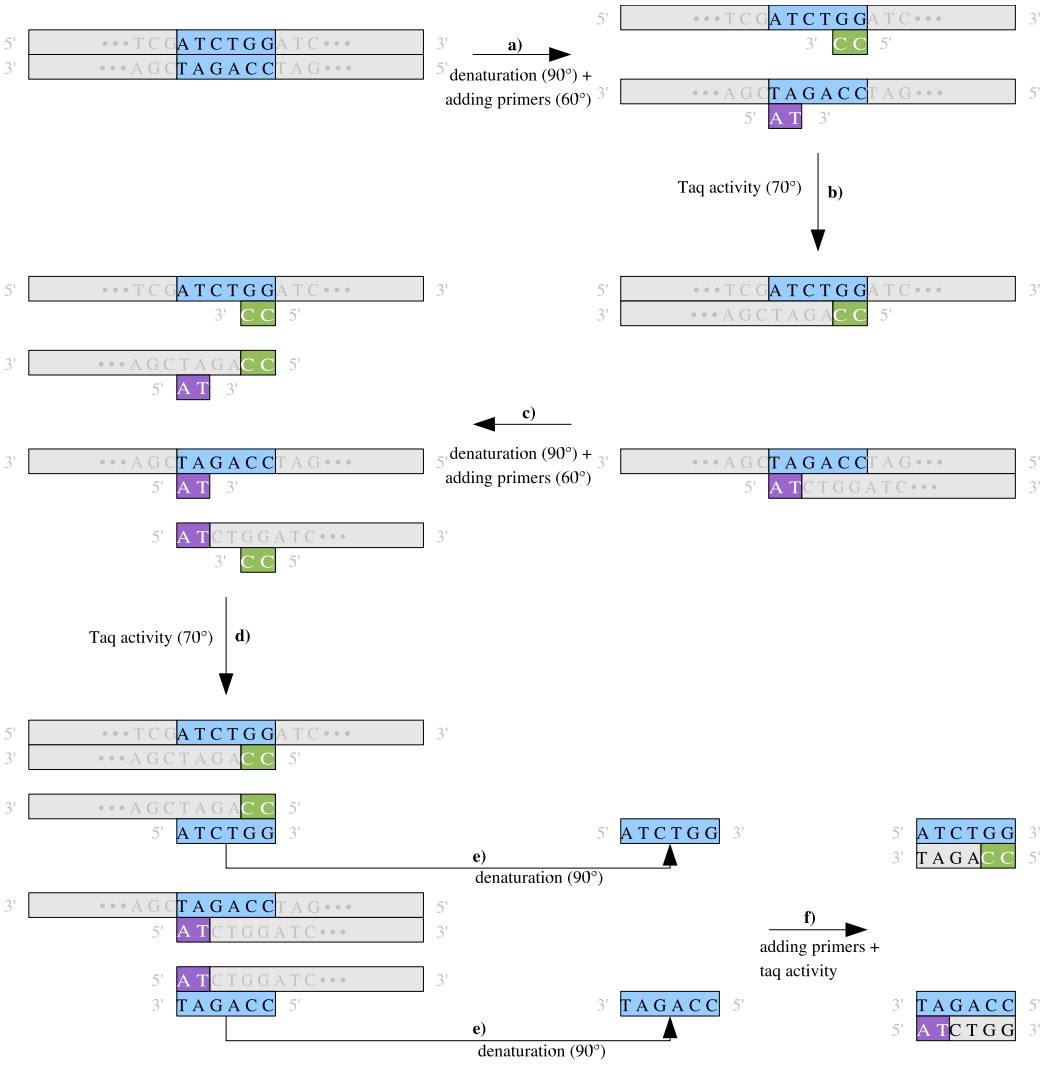

Figure B.1: Steps carried out during polymerase chain reaction (see text for details). Upper left: double stranded DNA molecule. The target fragment to be copied by means of PCR is highlighted in blue. a) Denaturation of DNA and adding of primers (purple and green). b) Taq polymerase reproduces sequences starting from primer in 5' to 3' direction. c) Second denaturation and adding of primers. d) Second elongation by Taq polymerase. Note that both target strands are reproduced. e) Third denaturation leads to release of the target strands (other molecules not shown for clarity). f) Adding of primers and Taq leads to two copies of the desired fragment.

Before the onset of the PCR process, DNA is put into a vial together with the primers and Taq polymerase as well as with a supply of the nucleotides A, G, C and T which will be needed to synthesize complementary DNA strands later on. In the first step of PCR (step a) in Fig B.1) this vial is heated up to 90 ℃ for 30 seconds such that the DNA in it denaturates and finally disintegrates into its two complementary strands. After it is slowly cooled down to 60 ℃ the primers (highlighted in purple and green, respectively, in Fig. B.1) in the vial start to base pair with some of the nucleotides at the 3' ends of each of the two target strands.

The vial is then heated to 70 ℃ (step b) in Fig. B.1) thus promoting Taq polymerase activity. Hereupon Taq polymerase starts to add the nucleotides complementary to those found in the original DNA molecule to the 3' end of the primer. The result of this synthesis are two partly double stranded DNA molecules whereas the nucleotides downstream of the 3' end of the target fragment remain unpaired.

In the next step (step c) in Fig. B.1), the vial is heated to 90 ℃ again such that the partly double stranded DNA molecules fall apart. Afterwards, the vial is cooled down to 60 ℃ allowing primers to bind again.

Hereon the temperature is increased to 70 ℃ (step d) in Fig. B.1) leading to Taq polymerase activity and hence to another set of partly double stranded DNA molecules. Thereby, in two of the four obtained molecules (the second and the fourth in Fig. B.1 d) the shorter of the two strands matches exactly one of the two target strands. These short strands are then separated from their partner strands by another heating to 90 ℃ (step e) in Fig. B.1). Thus, the upper and the lower target strand are reproduced. Starting from

these two (separated) target strands two complete target fragments can be synthesized. For this purpose, the vial is cooled down to 60 ℃ such that the primers can bind to the 3' ends of the copies of the target strands. Finally, the vial is heated to 70 ℃ again leading to the build up of the complementary strands by Taq polymerase (both steps summarised in step f) of Fig. B.1) such that eventually two copies of the target fragment are obtained.

To produce further copies of this fragment only three steps are needed in the further course of PCR. First, the vial has to be heated to 90 ℃ in order to disintegrate each of the double stranded copies of the target fragment into two single strands. Second, the vial needs to be cooled down such that the primers can bind to these single strands. Finally, the temperature has to be increased such that Taq polymerase adds the complementary strands to the single strands. These steps are repeated until the desired number of target fragments is obtained. Note, that thereby the number of copied fragments increases exponentially.

To be used in the production of spotted microarrays the two strands of the target fragments synthesized by PCR need to be separated. This is achieved by a final heating to 90 ℃ at the end of the PCR procedure upon which certain additives are added to the vial which block renaturation of DNA at lower temperatures. A multitude of such encymes has been developed up to now and a short overview of them can be found, for instance, in [92].

Which kind of ssDNA is produced during PCR depends on the used primers which are usually provided by commercial vendors. Nowadays, primers of some dozens of bases in length can be produced containing any desired sequence. With them, virtually any fragment of the original DNA molecule

can be reproduced as long as it does not exceed a critical length of about 5000 base pairs. However, ssDNA fragments of maximally 1200 base pairs in length are used in microarrays.

Bibliography

[1] MIT Center for Biological and Computation Learning. CBCL face database #1. http://cbcl.mit.edu/cbcl/software-datasets/FaceData2.html.

[2] AT&T Laboratories Cambridge. The ORL database of faces. http://www.uk.research.att.com/facedatabase.html.

[3] Genechip® expression analysis. *Technical manual*, Affymetrix®. 2004, http://www.affymetrix.com/support/technical/manual/expression%_manual.affx.

[4] Genechip® arrays provide optimal sensitivity and specificity for microarray expression analysis. *Technical note*, Affymetrix®. April 2006, http://www.affymetrix.com/support/technical/technotes/25mer_t%echnote.pdf.

[5] Genechip® expression analysis - data analysis fundamentals. *Technical manual*, Affymetrix®. 2006, http://www.affymetrix.com/support/technical/manual/expression%_manual.affx.

[6] Manufacturing of genechip® microarrays and building models. *Lesson plan*, Affymetrix®. April 2006, http://www.affymetrix.com/corporate/outreach/lesson_plan/down%loads/student_manual_activities/activity3/.

[7] Statistical algorithms description document. *White paper*, Affymetrix®. 2006, http://www.affymetrix.com/support/technical/whitepapers/sadd_%whitepaper.pdf.

[8] C. Bauer. Independent Component Analysis of Biomedical Signals. Modern Data Analysis Techniques Can Assist Medical Decision Making. Ph.D. thesis, Universidad de Regensburg (Alemania). Tesis Doctoral. 2001.

[9] A. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, pages 1129–1159. 1995.

[10] A. Bell and T.J. Sejnowski. Fast blind separation based on information theory. *Proceedings of International Symposium on Non-Linear Theory and Applications*, 1:43–47. NTA Research Society of IEICE. 1995.

[11] Douglas L. Black. Protein diversity from alternative splicing: A challenge for bioinformatics and post-genome biology. *Cell*, 103:367–370. October 2000.

[12] S. Bozinoski and H. L. Nguyen Thi. Séparation de sources á bande large dans un mélange convolutif. *Proceedings of Ecole des Technique Avancées en Signal Image Parole*, 37(23):3327–3338. 1997.

[13] A. S. Bregman. Auditory Scene Analysis: The Perceptual Organization of Sound. MIT Press. 1990.

[14] Ali H. Brivanlou and James E. Darnell jr. Signal transduction and the control of gene expression. *Science*, 295:813–818. February 2002.

[15] Michael S. Brown, Jin Ye, and Robert B. Rawson. Regulated intramembrane proteolysis: A control mechanism conserved from bacteria to humans. *Cell*, 100:391–398. February 2000.

[16] Jennifer E. F. Buttler and James T. Kadonaga. The RNA, polymerase II, core promoter: a key component in the regulation of gene expression. *Genes and Development*, 16:2583–2592. 2002.

[17] J. F. Cardoso. Blind signal separation: Statistical principles. *Proceedings of the IEEE*, 86(10):2009–2025. 1998.

[18] Zheng Chen and James L. Manley. Core promoter elements and TAF,s contribute to the diversity of transcriptional activation in vertebrates. *Molecular and Cellular Biology*, 23(20):7350–7362. 2003.

[19] P. Chevalier. Les statistiques d'ordre supérieur en traitement d'antenne. *Proceedings of Ecole des Technicques Avancées en Signal Image Parole*, pages 237–243. 1996.

[20] P. Chevalier, V. Capdeville, and P. Comon. Performance of ho blind source separation methods: experimental results on ionospheric hf links. *Proceedings of the International Workshop on Independent Component Analysis*, pages 443–448. 1999.

[21] A. Cichocki and S. Amari. Adaptive Blind Signal And Image Processing. John Wiley, New York. New revised and improved edition. 2003.

[22] A. Cichocki, S. I. Amari, M. Adachi, and W. Kasprzak. Self-adaptive neural networks for blind separation of sources. *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2:157–160. May 1996.

[23] A. Cichocki, W. Kasprzak, and S. I. Amari. Multi-layer neural networks with a local adaptive learning rule for blind separation of source signals. *Proceedings of 1995 International Symposium on Non-linear Theory and Applications*, 1:61–65. 1995.

[24] Bradley Coe and Christine Antler. Spot your genes - an overview of the microarray. April 2006, `http://www.bioteach.ubc.ca/MolecularBiology/microarray/index.%htm\#ref2`.

[25] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314. 1994.

[26] The Gene Ontology Consortium. Gene ontology: Tool for the unification of biology. *Nature Genetics*, 25:25–29. 2000.

[27] Albert J. Courey and Songtao Jia. Transcriptional repression: the long and the short of it. *Genes and Development*, 15:2786–2796. 2001.

[28] S. Cruces, A. Cichocki, and L. Castedo. An unified perspective of blind source separation algorithms. *Actas de Learning 98*. 1998.

[29] Y. Deville and L. Andry. Application of blind source separation techniques to multi-tag contactless identification systems. *Proceedings of 1995 International Symposium on Non-linear Theory and Applications*, 1:73–78. 1995.

[30] Y. Deville, J. Damour, and N. Charkani. Improved multi-tag radio-frequency identification systems based on new source separation neural networks. *Proceedings of the International Workshop on Independent Component Analysis*, pages 449–454. 1999.

[31] Sorin Drăghici. Data Analysis Tools for DNA Microarrays. Chapman & Hall/CRC, London. 2003.

[32] E. W. Lang F. J. Theis. Geometric overcomplete ica. *Proceedings of teh 10th European Symposium on Artificial Neural Networks*, pages 217–222. 2002.

[33] M. Feng and K. D. Kammeyer. Application of source separation algorithms for mobile communication environment. *Proceedings of the International Workshop on Independent Component Analysis*, pages 431–436. 1999.

[34] Stanford: functional genomics facility. The microarray core facility. April 2006, `http://microarray.org/sfgf/`.

[35] G. Gelle, M. Colas, and G. Delaunay. Separation of convolutive mixtures of harmonic signals with a temporal approach. application to rotating machine monitoring. *Proceedings of the International Workshop on Independent Component Analysis*, pages 109–114. 1999.

[36] M. S. Gupta. Electrical noise: fundamentals and sources. IEEE Press. 1977.

[37] S. Handel. Listening: An Introduction to the Perception of Auditory Events. MIT Press. 1989.

[38] S. Haykin. Neural Networks. ISBN 0132733501. Prentince Hall, Upper Saddle River, NJ, USA. 1999.

[39] P. O. Hoyer and A. Hyvärinen. Independent component analysis applied to feature extraction from color and stereo images. *Network: Computation in Neural Systems*, 11(3):191–200. 2000.

[40] Patrik O. Hoyer. Non-negative sparse coding. *Neural Networks for Signal Processing XII (Proc. IEEE Workshop on Neural Networks for Signal Processing)*, pages 557–565. 2002.

[41] Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469. 2004.

[42] A. Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128. 1999.

[43] A. Hyvärinen, P. Hoyer, and E. Oja. Denoising of nongaussian data by independent component analysis and sparse coding. *Proceedings of the International Workshop on Independent Component Analysis*, pages 485–490. 1999.

[44] Aapo Hyvärinen. Fast and robust fixed-point algorithmus for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634. 1999.

[45] National Human Genome Research Institute. Dna microarray technology. April 2006, `http://www.genome.gov/10000533`.

[46] C. Jutten and J. Karhunen. Advances in nonlinear blind source separation. *Proceedings of the 4th International Workshop on Independent Component Analysis*, pages 245–256. 2003.

[47] J. Karhunen. Neural approaches to independent component analysis and source separation. *Proceedings of the 4th European Symposium on Artificial Neural Networks*, pages 249–266. 1996.

[48] I. Kopriva and A. Persin. Blind separation of optical tracker responses into independent components discriminates optical sources. *Proceedings of the International Workshop on Independent Component Analysis*, pages 31–36. 1999.

[49] J. L. Lacoume. A survey of source separation. *Proceedings of the International Workshop on Independent Component Analysis*, pages 1–6. January 1999.

[50] J. L. Lacoume, P. O. Amblard, and P. Comon. Statistiques d'ordre supérieur pour le traitement du signal. Masson. January 1997.

[51] D. D. Lee and H. S. Seung. Learning of the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791. 1999.

[52] Daniel D. Lee and H. Sebastion Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791. 1999.

[53] S. I. Lee and S. Batzoglou. Ica-based clustering of genes from microarray expression data. *Advances in Neural Information Processing Systems*. December 2003.

[54] T. W. Lee. Independent component analysis: theory and applications. Kluwer Academic Publishers. 1998.

[55] Kei Lida, Motoaki Seki, Tetsuya Sakurai, Masakzou Satou, Kenji Akiyama, Tetsuro Toyoda, Akihiko Konagaya, and Kazou Shinozaki. Genome-wide analysis of alternative pre-mrna splicing in arabidopsis thaliana based on full-length cdna sequences. *Nucleic Acids Research*, 32(17):5096–5103. 2004.

[56] C.-P. Lui, P. W. Tucker, F. Mushinski, and F. R. Blattner. Mapping of heavy chain genes for mouse immunoglobulins M, and G. *Science*, 209:1348–1353. 1980.

[57] S. Makeig, T.P. Jung, A.J. Bell, and T.J. Sejnowski. Blind separation of auditory event-related brain responses into independent components. *Proceedings of National Academy of Sciences*, 98:10979–10984. 1998.

[58] Edio Maldonado, Michael Hampsey, and Danny Reinberg. Repression: Targeting the heart of the matter. *Cell*, 99:455–458. November 1999.

[59] David J. Mangelsdorf, Carl Thummel, Miguel Beato, Peter Herrlich, Günther Schütz, Kazuhiko Umeasono, Bruce Blumberg, Philippe Kast-

ner, Manuel Mark, Pierre Chambon, and Ronald M. Evans. The nuclear receptor superfamily: The second decade. *Cell*, 83:835–839. December 1995.

[60] H. Marsmann. A neural net approach to the source separation problem. Master's thesis, Universidad de Twente, Holland. PhD thesis. 1995.

[61] Goeffrey J. McLachlan, Kim-Anh Do, and Christophe Amboise. Analyizing Microarray Gene Expression Data. Wiley Series in Probability and Statistics. John Wiley & Sons Inc., Hobokan, New Jersey. 2004.

[62] Menie Merika and Dimitris Thanos. Enhanceosomes. *Current Opinion in Genetics and Development*, 11:205–208. 2001.

[63] Jennifer Michalowski. Alternative splicing. *Howard Hughes Medical Institute Bulletin*, 18(2):23–28. September 2005.

[64] S. K. Mitra and J. F. Kaiser. Handbook for digital signal processing. John Wiley & Sons. 1993.

[65] Harvey Motulsky. Intuitive Biostatistics. ISBN: 0195086074. Oxford University Press, New York. October 1995.

[66] Asoke K. Nandi (editor). Blind Estimation Using Higher Order Statistics. ISBN 0-7923-8442-3. Kluwer Academic Publishers, Amsterdam, Netherlands. January 1999.

[67] H. L. Nguyen and C. Jutten. Blind source separation for convolutive mixtures. *Signal Processing*, 45:209–229. 1995.

[68] D. Nuzillard and J. N. Nuzillard. Blind source separation applied to non-orthogonal signal. *Proceedings of the International Workshop on Independent Component Analysis*, pages 5–30. 1999.

[69] R.M. Olalla. Sistema Híbrido de Cancelación de Ruido en Señales de Voz Basado en Filtrado Adaptativo y Sustracción Espectral. Ph.D. thesis, Departamento de Arquitecctura y Tecnología de Sistemas Informáticos. Facultad de Informática. Universidad Politécnica de Madrid. 2002.

[70] A. ParaschivIonescu, C. Jutten, A. M. Ionescu, A. Chovet, and A. Rusu. High performance magnetic field smart sensor arrays with source separation. *Proceedings of the First International Conference on Modelling and Simulation of Microsystems*. 1998.

[71] L. Parra, K. R. Mueller, C. Spence, A. Ziehe, and P. Sajda. Unmixing hyperspectral data. *In Advances in Neural Information Processing Systems*, pages 942–948. 2000.

[72] M. D. Plumbley. Algorithms for nonnegative independent component anlaysis. *IEEE Trans. Neural Networks*, 14(3):534–543. May 2003.

[73] A. Prieto, C. G. Puntonet, B. Brieto, and M. Rodríguez. A competitive neural network for blind source separation of sources based on geometric properties. *Lecture Notes in Computer Science*, 1240. June 1997.

[74] B. Prieto. Nuevos Algoritmos de Separación de Fuentes en Medios Lineales. Master's thesis, Universidad de Granada (Spain). Tesis doctoral. 1999.

[75] C. G. Puntonet, M. R. Álvarez, A. Prieto, and B. Prieto. Separation of speech signals for nonlinear mixtures. *Lecture Notes in Computer Science*, 1607:665–673. 1999.

[76] C. G. Puntonet, A. Mansour, and C. Jutten. Un algorithme géomeétrique pour la séparation de sources. *Proceedings of the 15éme Colloque GRETSI*. September 1995.

[77] C. G. Puntonet, A. Prieto, C. Jutten, M. Rodriguez-Alvarez, and J. Ortega. Separation of sources: a geometry-based procedure for reconstruction of n-valued signals. *Signal Processing*, 46(3):267–284. 1995.

[78] T. Ristaniemi and J. Joutsensalo. On the performance of blind source separation in cdma downlink. *Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation*, pages 437–442. 1999.

[79] M. Rodirguez and C. G. Puntonet. A new geometry-based procedure for blind source separation. *Proceedings of the 2nd International ISCS Symposium on Engineering of Intelligent Systems*. Junio 2000.

[80] M. Rodriguez, C.G. Puntonet, L. Parrilla, and A. Díaz. Redes neuronales aplicadas a separación ciega de se nales. *Actas del Seminario Anual de Automática, Electrónica e Instrumentatción*, 1998. June.

[81] M. Rodríguez, C.G. Puntonet, and I. Rojas. Separation of sources based on the partitioning of the space of observations. *Lecture Notes in Computer Science, Springer Verlag*, (2085):762–769. 2001.

[82] H. Sahlin and H. Broman. Blind separation of images. *roceedings of the 30th ASILOMAR Conference on Signals, Systems and Computers*. 1996.

[83] I. Schief, M. Stetter, J. E Mayhew, S. Askew, N. Mc. Loughlin, J. B. Levitt, S. Lund, and K. Obermaye. Blind separation of spatial signal patterns from optical imaging records. *Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation*, pages 179–184. 1999.

[84] Dietmar Schmucker, James C. Clemens, Huidy Shu, Carolyn A. Worby, Jian Xiao, Marco Muda, and S. Lawrence Zipursky. Drosophila Dscam, is an axon guidance receptor exhibiting extraordinary molecular diversity. *Cell*, 101:671–684. June 2000.

[85] M. Soltysik-Espanola, D. C. Klinzing, K. Pfarr, R. D. Burke, and G. Ernst. Endo16, a large multidomain protein found on the surface and ECM, of endodermal cells during sea urchin gastrulation, binds calcium. *Developmental biology*, 165:73–85. 1994.

[86] Kurt Stadlthanner, Fabian J. Theis, Carlos G. Puntonet, and Elmar W. Lang. Extended sparse nonnegative matrix factorization. *Lecture Notes in Computer Science*, 3512:249–257. 2005.

[87] Yuichiro Takagi and Roger D. Kornberg. Mediator as a general transcription factor. *The Journal of Biological Chemistry*, 281(1):80–89. January 2006.

[88] A. Taleb, C. Jutten, and S. Olympief. Source separation in post nonlinear mixtures: an entropy-based algorithm. *Proceedings of the International Conference on Acoustic and Speech Signal Processing*. May 1999.

[89] F. J. Theis, E. W. Lang, Tobias Westenhuber, and G. C. Puntonet. Overcomplete bss with a geometric algorithm. *Proceedings of International Conference on Artificial Neural Networks*. 2002.

[90] N. Thirion. Séparation d'ondes en prospection sismique. Tesis doctoral, Instituto Nacional Politécnico de Grenoble (Francia). 1995.

[91] K. Torkolla. Blind separation for audio signals - are we there yet? *Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation*, pages 239–244. 1999.

[92] David W. Ussery. The Encyclopedia of Genetics, volume 1, chapter DNA Denaturation, pages 550–553. Academic Press, New York. 2001.

[93] S. Van-Dijk and M.Berg D. Thierens. Scalability and efficiency of genetic algorithms for geometrical applications. *Proceedings of the Seventh International Conference on Parallel Problem Solving from Nature*, 1917:683–692. 2000.

[94] R. Vetter, J. M. Vesin, P. Celka, and U. Scherrer. Observer of the autonomic cardiac outflow in humans using non-casual blind source separation. *Proceedings of the International Workshop on Independent Component Analysis*, pages 161–166. 1999.

[95] R. Vigario, J. Sarela, V. Jousmaki, and E. Oja. Independent component anlysis in decomposition of auditory and somatosensory evoked fields. *Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation*, pages 167–172. 1999.

[96] P.J. Werbos. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Phd thesis, Harvard University. 1974.

[97] Jumming Yie, Stanley Liang, Mente Merika, and Dimitirs Thanos. Intra- and intermolecular cooperative binding of high-mobility-group protein I(Y), to the beta-interferon promoter. *Molecular and Cellular Biology*, 17(7):3649–3662. July 1997.

[98] A. Ypma and P. Pajunen. Rotating machine vibration analysis with second-order independent component analysis. *roceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation*, pages 37–42. 1999.

[99] Chiou-Hwa Yuh, Hamid Bolouri, and Eric H. Davidson. Genomic cis-regulatory logic: Experimental and computational analysis of a sea urchin gene. *Science*, 279:1896–1902. March 1998.