TREBALL DE FI DE MÀSTER

Màster en Estadística i Investigació Operativa

Títol: Kernel PCA per a l'anàlisi de dades òmiques

Autor: Núria Planell Picola

Director: Esteban Vegas Lozano

Departament: Estadística (Universitat de Barcelona)

Convocatòria: Juny 2015

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

Universitat Politècnica de Catalunya
Facultat de Matemàtiques i Estadística

Treball fi de màster

Kernel PCA per a l'anàlisi de dades òmiques

Núria Planell Picola

Director: Esteban Vegas Lozano

Departament d'Estadística - Facultat de Biologia - Universitat de Barcelona

Resum

Un organisme és un sistema biològic format per subsistemes que interaccionen de forma coordinada. Els avenços tecnològics dels darrers anys han permès l'estudi de cada un dels subsistemes donant lloc a les òmiques: genòmica, transcriptòmica, proteòmica, etc. Amb la inquietud de modelar aquest sistema, neix la necessitat d'integrar les dades provinents de les diferentes òmiques, presentant-se no tan sols com un repte conceptual, sinó com un obstacle en l'anàlisi diari de dades òmiques.

Les funcions *kernel*, per les seves propietats, permeten combinar diferents tipus de dades, essent una possible via per a la integració de dades òmiques. Amb l'objectiu d'explorar l'ús d'aquestes, s'estudien els fonaments estadístics de l'anàlisi de components principals basat en funcions *kernel* (*kernel* PCA), es proporcionen eines que permeten integrar conjunts de dades, representar variables originals i cercar variables d'interès en el *kernel* PCA i, finalment, s'inspecciona un conjunt de dades transcripcionals de pacients amb colitis ulcerosa utilitzant les eines desenvolupades.

Les eines creades s'han implementat en R. Per una banda s'ha desenvolupat la funció `KPCAplus`, que permet analitzar i integrar diferents conjunts de dades, i per altra banda la funció `KPCAplusGUI`, que executa la versió web de `KPCAplus`.

L'anàlisi de les dades de pacients amb colitis ulcerosa, prenent com a referència els resultats publicats, ens han permès validar la representació de variables en el *kernel* PCA i explorar la cerca de noves variables.

Paraules clau: funcions *kernel*, *kernel* PCA, òmiques, integració de dades, colitis ulcerosa, R

Abstract

An organism is a biologic system composed of subsystems that interact in a coordinated way. Technological advances have facilitated the study of these subsystems, most recently by use to the omics: genomics, transcriptomics, proteomics, etc. Increasing interest in modeling biological systems has required the integration of data from different omics sources. This has become not only a conceptual challenge, but also a practical problem in routine data analysis.

The functional properties of the kernel allows for the combination of different data types, providing a possible method for omics data integration. To fully explore these functions using omics data analysis, the statistical foundations of principal component analysis, based on kernel functions (kernel PCA), are described herein. In-house tools for data integration, original variable representation and new variable discovery are provided. Finally, a transcriptional dataset from ulcerative colitis patients is investigated using these tools.

The statistical software R was used to develop the tools. A function for analyzing and integrating different data sets was created, designated `KPCAplus`. Moreover, another function called `KPCAplusGUI`, which runs a graphic user interface from `KPCAplus`, was created.

Ulcerative colitis data analysis, taking the published results as a reference, enables us to validate the variables representation in the kernel PCA and further explore the new variable discovery strategy.

Keywords: kernel function, kernel PCA, omics, data integration, ulcerative colitis, R

Continguts

Notacions

x	Valor escalar
$\mathbf{x}$	Vector columna
$\mathbf{x}^T$	Vector transposat
$\|\mathbf{x}\|$	Norma del vector $\mathbf{x}$
$\mathbf{X}$	Matriu
n	Nombre d'observacions
p	Nombre de variables
$\mathbb{R}$	Espai dels reals
$\mathcal{X}$	Espai d'entrada o *input space*
$\mathcal{F}$	Espai de característiques o *feature space*
$\phi(\mathbf{x})$	Funció que passa $\mathbf{x}$ de $\mathcal{X}$ a $\mathcal{F}$
$k(\mathbf{x}, \mathbf{x}')$	Funció *kernel*
$\mathbf{K}$	Matriu *kernel*
$\mathbf{I}$	Matriu identitat
$\mathbf{1}_p$	Vector d'uns de llargada p (veure pàgina 29)

Introducció

Motivació

L'individu, del llatí *individuus*, que es refereix a allò que no es pot dividir, representa la unitat mínima i no divisible d'un sistema. Aquest sistema, format per grups de subsistemes que treballen de forma coordinada, dóna lloc a una complexa xarxa de components i interaccions que parteix del genoma per esdevenir en un fenotip (Figura 1).

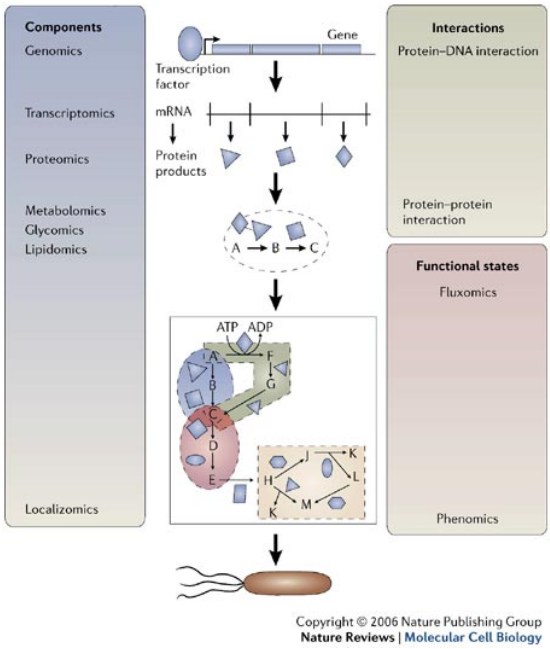

Figura 1: Representació de la complexa xarxa de components i interaccions d'un organisme unicel·lular (*Escherichia coli*). El fenotip cel·lular en un moment i condicions determinades d'aquest organisme vindrà definit pel ADN (genòmica) que es transcriu a RNAm (transcriptòmica) i seguidament es tradueix a proteïnes (proteòmica). Les proteïnes, involucrades en la catalització de reaccions químiques del metabolisme generant metabòlits (metabolòmica), glicoproteïnes (glicòmica) i lípids (lipidòmica), entre altres; dirigiran el comportament cel·lular cap a un fenotip o altre [1].

Els avenços tecnològics dels darrers anys han permès l'estudi dels components de cada un dels subsistemes, donant lloc a les òmiques: genòmica, epigenòmica, transcriptòmica, proteòmica, metabolòmica, glicòmica i lipidòmica, entre altres, permetent fer un *escàner*, a diferents nivells, d'un individu en un moment i unes condicions determinades. L'anàlisi individual de cada una de les òmiques és la base de gran part dels estudis actuals; no obstant, davant de la gran quantitat de dades (procedents de diverses òmiques) que es poden generar per a un únic individu, i amb la inquietud d'obtenir una millor caracterització i comprensió dels sistemes biològics, neix la necessitat d'integrar tota aquesta informació [1–5].

Aquesta inquietud per a la integració de dades està esdevenint una necessitat real en el meu àmbit professional. Des del 2008 formo part del grup d'investigació en malaltia inflamatòria intestinal de l'Hospital Clínic i Provincial de Barcelona, en el que he pogut participar en el desenvolupament de varis projectes transcripcionals relacionats amb la malaltia. No obstant, els projectes que s'estan realitzant actualment, molt més ambiciosos, són un repte conceptual i metodològic, en els que ens haurem d'enfrontar a un gran nombre de dades procedents de diferents òmiques.

Una de les aproximacions possibles per a la integració de dades es centra en l'ús de mètodes que es basen en la transformació de les dades (*transformation-based integraton*). Aquests mètodes transformen les dades d'entrada mitjançant funcions *kernel* o *graphs* permetent combinar-les per tal d'obtenir un únic conjunt de dades, però no hem d'oblidar que són mètodes complicats d'interpretar i amb els que s'haurà d'anar amb molt de compte per no perdre les característiques pròpies de les dades [2].

El departament d'Estadística de la Universitat de Barcelona (UB) està investigant ampliament aquests mètodes, entre els que trobem l'estudi del *kernel* PCA, on busquen afrontar la manca d'interpretabilitat d'aquest i explorar el seu ús en la integració de dades [6].

Conseqüentment, el projecte que es presenta a continuació es planteja amb la finalitat d'explorar i proporcionar eines per a l'ús del *kernel* PCA en la integració i l'anàlisi de dades òmiques. Per tal d'avaluar de forma crítica l'ús i interpretabil-

itat d'aquest mètode, es fa una primera aproximació en un únic conjunt de dades transcripcionals conegudes.

Objectius

Els objectius principals del projecte són:

- Revisió de les funcions *kernel* i l'anàlisi de components principals basat en funcions *kernel*. Amb especial èmfasi en la integració de dades i en les noves metodologies descrites per a la visualització i cerca de variables.
- Implementació en R d'una funció per a l'extensió del *kernel* PCA habitual, facilitant la integració de diferents conjunts de dades, la representació de les variables originals, la representació de combinacions de variables i la cerca de noves variables.
- Creació d'una aplicació web per a la funció R creada amb la corresponent documentació en HTML.
- Aplicació del *kernel* PCA en un conjunt de dades reals conegutes.

La memòria del projecte segueix la següent estructura: en el Capítol 1 es donen els coneixements estadístics bàsics, on es descriuen les bases de les funcions *kernel* i del *kernel* PCA; en el Capítol 2 es descriu la implementació en R d'una funció per a la integració i visualització de dades en el *kernel* PCA, presentant també la seva aplicació web i el paquet en R resultant; i en el Capítol 3 s'exemplifica l'ús de l'estratègia descrita en un conjunt de dades reals. Finalment, es presenten les conclusions del projecte i les possibles vies futures d'estudi.

Capítol 1.
Mètodes basats en *kernels*

1. Aprenentatge estadístic

La cerca i detecció de regularitats, relacions, pautes o patrons en conjunts de dades és l'objectiu del que es coneix com aprenentatge estadístic. Aquesta disciplina apareix de forma natural tan bon punt es disposa d'un conjunt de dades, convertint-se en una eina bàsica per a determinar el funcionament del món que ens envolta.
Dins l'aprenentatge estadístic es diferencia entre l'aprenentatge supervisat i el no supervisat [7].

- **Aprenentatge supervisat**. L'aprenentatge supervisat consisteix en establir un mecanisme de classificació o regressió a partir d'unes dades d'entrenament amb el valor de la variable d'interès conegut. Un cop establert el mecanisme predictor, s'avalua la capacitat d'aquest en un conjunt de dades de prova. Els mètodes supervisats engloben la regressió lineal, la regressió logística, l'anàlisi discriminant (*Linear Discriminant Analysis*, LDA), els arbres de decisió i les màquines de vector de suport (*Support Vector Machines*, SVM), entre altres.

- **Aprenentatge no supervisat**. L'aprenentatge no supervisat es caracteritza per la manca d'una variable d'interès en el conjunt de dades. Les dades no presenten una estructura prèviament establerta, pel que l'objectiu d'aquests tipus d'anàlisis consisteix en detectar l'existència de determinades estructures en el conjunt de dades. Dins dels mètodes no supervisats trobem l'anàlisi de components principals (*Principal Components Analysis*, PCA) i els mètodes de clusterització de dades (*Clustering Methods*).

L'ús d'aquests mètodes en estudis de dades òmiques és molt comú. Alguns exemples clars i habituals podrien ser l'ús de mètodes de regressió per identificar variables predictores d'un fenotip d'interès (per exemple, la detecció d'un SNP que s'associï a una malaltia determinada), o l'anàlisi de components principals per a visualitzar la distribució dels individus o les variables d'un conjunt de dades [8]. No obstant, la diversitat en la naturalesa de les dades ha portat al desenvolupament de mètodes més versàtils que permetin processar, analitzar i comparar molts tipus de dades diferents, com els **mètodes basats en *kernels***. Tot i que els orígens d'aquests mètodes en l'estadística daten del 1950, l'ús en les òmiques és relativament recent [9, 10].

2. Mètodes basats en *kernels*

Els mètodes basats en *kernels* o mètodes *kernel*, dins d'un marc modular, ens permeten construir versions no lineals d'algoritmes d'aprenentatge lineals, permetent detectar relacions més complexes que les identificades amb els mètodes lineals habituals.

Partint d'un conjunt de dades $\mathcal{X}$, la naturalesa del qual pot ser molt variada (vectors, textos, imatges, etc.), s'aplica una funció ϕ, anomenada *embedding*

$$\phi : \mathcal{X} \to \mathcal{F}$$

de manera que cada element de l'espai $\mathcal{X}$ es projectarà en un nou espai $\mathcal{F}$ (espai de característiques) sobre el que es podran aplicar els mètodes de detecció de patrons lineals. Així, la detecció de patrons en $\mathcal{F}$ ens estarà donant patrons no lineals de l'espai d'entrada $\mathcal{X}$ (Figura 2) [11].

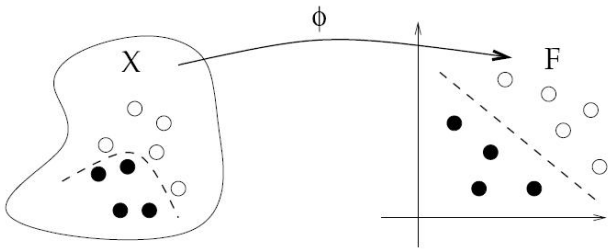

Figura 2: Il·lustració de la projecció ϕ. A l'esquerra es representa el conjunt de dades d'entrada $\mathcal{X}$ i a la dreta les dades a l'espai de característiques $\mathcal{F}$, mostrant com després d'aplicar la funció *embedding* les dades són linealment separables en $\mathcal{F}$ [11].

Per a evitar el càlcul de les imatges ϕ dels elements de $\mathcal{X}$ i la cerca de patrons en espais $\mathcal{F}$ de grans dimensions s'ha recorregut a l'estratègia descrita com *kernel trick*.

El *kernel trick* es basa en l'existència de les funcions *kernel* juntament amb la *kernelització* dels algoritmes de cerca de patrons per evitar transitar per la funció ϕ convertint-se, així, en la base dels mètodes basats en *kernels* (Figura 3).

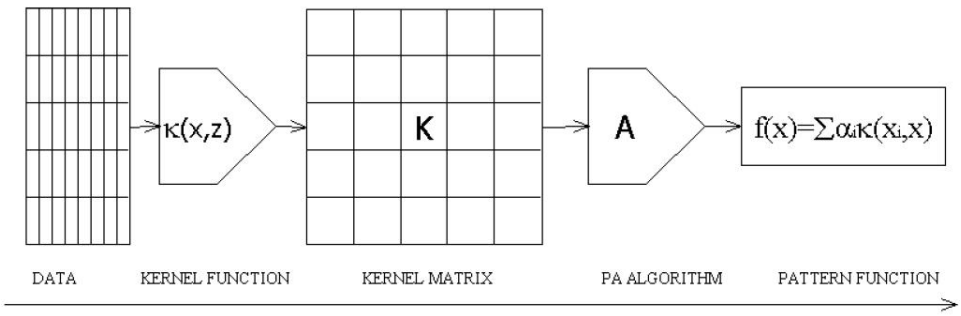

Figura 3: Esquematització dels mètodes basats en *kernels* per a l'anàlisi de reconeixement de patrons. A partir d'un conjunt de dades i mitjançant una funció *kernel* s'obté la matriu *kernel*. Sobre la matriu *kernel* obtinguda es podran aplicar els diferents mètodes basats en *kernels* (*Pattern Analysis Algorithm*) per a la detecció de patrons [12].

Alguns mètodes en els que es poden aplicar funcions *kernel* són:

- Màquines de vector de suport (*Support Vector Machines*, SVM)
- Mètodes de clusterització de dades (*Clustering Methods*)
- Anàlisi de components principals (*Principal Components Analysis*, PCA)
- Anàlisi de correlació canònica (*Canonical-Correlation Analysis*, CCA)

que es caracteritzen per utilitzar el resultat d'una funció *kernel* (matriu *kernel*) com a informació d'entrada.

2.1. Funció *kernel*

La funció *kernel* és l'element principal dels mètodes basats en *kernel*.
Suposem que disposem d'un conjunt de dades

$$S = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \quad amb \quad \mathbf{x}_i \in \mathcal{X} \quad \forall\ i \in \{1, \ldots, n\} \quad i \quad \mathcal{X} \subseteq \mathbb{R}^{\mathrm{p}}$$

i que s'ha definit la funció ϕ (*embedding function*)

$$\begin{array}{rccc} \phi: & \mathcal{X} & \rightarrow & \mathcal{F} \\ & \mathbf{x} & \rightarrow & \phi(\mathbf{x}) \end{array}$$

que projecta els elements de S en un espai de Hilbert $\mathcal{F}$, anomenat espai de característiques (*feature space*), de dimensió superior P, $P > p$, podent ser infinita.

La funció *embedding* ϕ transforma el conjunt de dades original S (format per vectors p-dimensionals) en un nou conjunt de dades

$$\phi(S) = \{\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n)\}$$

format per vectors P-dimensionals. Els algoritmes de detecció de patrons podran treballar sobre aquest nou conjunt de dades $\phi(S)$ per identificar relacions lineals entre els elements $\phi(\mathbf{x}_i)$; no obstant, l'elevada dimensionalitat P de l'espai de característiques $\mathcal{F}$ apareix com un *handicap* en el procés.

Suposem que podem modificar els algoritmes de detecció de patrons de manera que per identificar patrons en $\mathcal{F}$ no sigui necessari conèixer les imatges dels *embedding* dels elements d'entrada $\phi(S)$ sinó que tan sols coneixent els productes escalars d'aquestes imatges (mòduls i posicions relatives en $\mathcal{F}$) sigui suficient per a l'ús d'aquests algoritmes. Quan aquesta aproximació sigui viable es dirà que l'algoritme pot ser *kernelitzat*.

Així, la *kernelització* ens facilita la cerca de patrons en $\mathcal{F}$, però encara necessitem definir la projecció (*embed*) de les dades d'entrada $\mathcal{X}$ a l'espai de característiques $\mathcal{F}$ per a poder calcular els productes escalars $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$.

Des d'aquest punt de vista, ens interessaria que fos possible determinar la funció $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ tal que ens permetés obtenir simultàniament:

- La transformació per *embedding* de les dades de S a $\phi(S)$
- El càlcul dels productes escalars de les imatges $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ amb $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X} \quad \forall\ i, j \in \{1, \ldots, n\}$

Definint aquesta funció per a cadascuna de les possibles parelles dels elements del conjunt S com

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$$

ens proporciona la informació necessària pels mètodes de detecció de patrons *kernelitzats*. La funció k es diu que és la **funció *kernel*** corresponent al *embedding*, $\phi : \mathcal{X} \to \mathcal{F}$.

En general, la funció *kernel* podrà determinar-se a partir de la funció *embedding* ϕ i del producte escalar de l'espai de característiques $\mathcal{F}$. No obstant, ens trobarem amb situacions on la dimensió de l'espai $\mathcal{F}$ pot ser infinita, resultant impossible trobar la funció *kernel*.

Per tant, el que es sol fer és treballar a la inversa. Donada una funció k, definida en $\mathcal{X} \times \mathcal{X}$ i amb valors reals, es determina un espai de caracterítiques $\mathcal{F}$ i un embedding $\phi : \mathcal{X} \to \mathcal{F}$ de manera que $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ per tots els elements de $\mathcal{X}$. Aquesta aproximació serà vàlida sempre que la funció *kernel* sigui simètrica i definida positiva.

Definició formal de *kernel*

Una funció $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ és una funció *kernel* si, i només si, és

- Simètrica $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x}) \quad \forall\ \mathbf{x}, \mathbf{x}' \in \mathcal{X}$
- Definida positiva

$$\sum_{i=1}^{n} \sum_{i=j}^{n} c_i c_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0$$

  per a qualsevol $n > 0$, qualsevol elecció de n objectes $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$ i qualsevol elecció de nombres reals $c_1, \ldots, c_n \in \mathbb{R}$

D'aquesta manera, mitjançant l'ús de la funció *kernel*, les dades no es representen individualment, sinó que es presenten com a comparacions 2 a 2, generant una matriu que s'anomena **matriu *kernel***. En comptes de definir $\phi : \mathcal{X} \to \mathcal{F}$ per representar cada element $\mathbf{x} \in \mathcal{X}$ com a $\phi(\mathbf{x}) \in \mathcal{F}$, s'utilitzen els valors reals de la funció *kernel* $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ per representar el conjunt de dades S en una matriu $\mathbf{K}$ $(n \times n)$ amb la comparativa aparellada $\mathbf{K}_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$ (Figura 4).

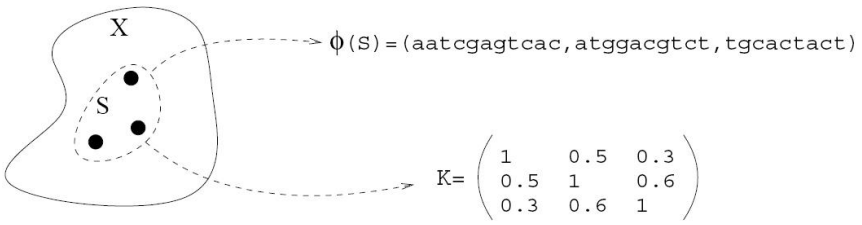

Figura 4: Dues formes diferents de representar el conjunt de dades S en l'espai de característiques $\mathcal{F}$. Es suposa que $\mathcal{X}$ és el conjunt de tots els oligonucleòtids, i S un conjunt de tres oligonucleòtids en particular. La forma clàssica de representar S consisteix en definir $\phi(\mathbf{x})$ per cada element $\mathbf{x} \in \mathcal{X}$ (part superior). Els mètodes *kernel*, gràcies a l'ús de les funcions *kernels*, es basen en una representació diferent de S; representen les dades com una matriu de similaritats entre els elements (matriu *kernel* ($\mathbf{K}$), part inferior) [11].

Per tant, l'elecció de la funció *kernel* serà equivalent a l'elecció de ϕ (*embedding*); essent la base dels mètodes *kernelitzats*.

Alguns dels *kernels* més comuns són [13] :

- *Kernel* lineal. El *kernel* lineal és el *kernel* més senzill i es defineix com el producte escalar de dos vectors

$$k(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle$$

  Molt usat quan es disposa de vectors amb dades molt disperses, però cal destacar que només es podran emprar quan les dades a analitzar siguin vectors.

- *Kernel* polinòmic. El *kernel* polinòmic es defineix generalment com

$$k(\mathbf{x}, \mathbf{x}') = (\alpha \cdot \langle \mathbf{x}, \mathbf{x}' \rangle + c)^d$$

on d és el grau del polinomi ($d > 0$), α és un paràmetre d'escala ($\alpha > 0$) i c és el *offset* ($c > 0$). Quan $d = 1$, $\alpha = 1$ i $c = 0$, el *kernel* polinòmic es redueix a un *kernel* lineal. Usat habitualment en classificació d'imatges.

- *Kernel* gaussià (*Gaussian radial basis (RGB)*). El *kernel* gaussià es defineix com

$$k(\mathbf{x}, \mathbf{x}') = exp\left(-\sigma||\mathbf{x} - \mathbf{x}'||^2\right)$$

on el paràmetre σ permet controlar la flexibilitat d'aquest *kernel*. Si el paràmetre σ és molt petit, la funció *kernel* assignarà valors molt propers a zero a les parelles formades pels elements de l'espai d'entrada, encara que les diferències siguin molt petites. En aquest cas la matriu *kernel* s'assemblarà molt a la matriu identitat i tindrem problemes d'*overfitting*. Per contra, si el valor de σ és molt elevat, ens trobarem que els valors assignats per la funció *kernel* seran molt pròxims a 1, de manera que ens trobarem davant d'una funció pràcticament constant i no serà útil per a la detecció de patrons.
La flexibilitat d'aquest *kernel* fa que sigui un dels més usats, sobretot quan no es té informació a priori de les dades.

- *Kernel* ANOVA. El *kernel* ANOVA, igual que el gaussià, es basa en una funció *kernel radial basis* i es defineix com

$$k(\mathbf{x}, \mathbf{x}') = \left(\sum_{k=1}^{p} exp\left(-\sigma\left(x_k - x'_k\right)^2\right)\right)^d$$

S'ha descrit per a l'ús en problemes de regressió multidimensional [14].

- *Kernels* per a cadenes de caràcters (*string kernel*). En termes generals, els *kernels* per a cadenes de caràcters es poden entendre com una mesura de similitud entre parelles de cadenes de caràcters. Donades dues cadenes de caràcters a i b, com més similars siguin ambdues més elevat serà el valor del *kernel* $k(a, b)$. Existeixen diferents tipus de *kernels* per a cadenes de caràcters diferenciant-se, principalment, pel mètode en el que s'analitzen les coincidències, com el *spectrum kernel* que considera el nombre exacte de coincidències entre les cadenes per n caràcters o el *mismatch kernel* que permet la incorporació de *gaps* [13].

La varietat de *kernels* portarà a que, depenent del que es vulgui analitzar, i tenint en compte les propietats de cada *kernel*, el més indicat sigui un o altre.

Una característica molt interessant de les funcions *kernel*, i rellevant dins de l'àmbit de la integració de dades, és la combinació de *kernels*. Recordant les propietats dels *kernels*, funcions simètriques i definides positives, es demostra que la combinació de funcions simètriques i definides positives mitjançant operacions que conservin aquestes propietats generen nous *kernels* vàlids.

Les operacions que es podran aplicar per a combinar *kernels* mantenint aquestes dues propietats són:

$$k(\mathbf{x},\mathbf{x}') = \begin{cases} k_1(\mathbf{x},\mathbf{x}') + k_2(\mathbf{x},\mathbf{x}') \\ k_1(\mathbf{x},\mathbf{x}')\, k_2(\mathbf{x},\mathbf{x}') \\ f(\mathbf{x})\, f(\mathbf{x}') \\ k_3(\phi(\mathbf{x}), \phi(\mathbf{x}')) \\ \mathbf{x}^T \mathbf{B} \mathbf{x}' \end{cases}$$

essent vàlides les funcions *kernel* obtingudes si, k_1 i k_2 són *kernels* en $\mathcal{X} \times \mathcal{X}$, amb $\mathcal{X} \subseteq \mathbb{R}^p$, $f(\cdot)$ una funció real definida en $\mathcal{X}$, $\phi : \mathcal{X} \to \mathbb{R}^P$, k_3 un *kernel* sobre $\mathbb{R}^P \times \mathbb{R}^P$ i $\mathbf{B}$ una matriu $p \times p$ simètrica i definida positiva.

2.2. Matriu *kernel*

La matriu *kernel* ($\mathbf{K}$) és el resultat directe d'aplicar una funció *kernel* al conjunt de dades d'entrada S. Aquesta matriu conté els productes escalars en $\mathcal{F}$ de cadascun dels individus, i es defineix com

$$\mathbf{K} = \begin{pmatrix} k(\mathbf{x}_1,\mathbf{x}_1) & k(\mathbf{x}_1,\mathbf{x}_2) & \cdots & k(\mathbf{x}_1,\mathbf{x}_n) \\ k(\mathbf{x}_2,\mathbf{x}_1) & k(\mathbf{x}_2,\mathbf{x}_2) & \cdots & k(\mathbf{x}_2,\mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_n,\mathbf{x}_1) & k(\mathbf{x}_n,\mathbf{x}_2) & \cdots & k(\mathbf{x}_n,\mathbf{x}_n) \end{pmatrix} = \begin{pmatrix} \langle\phi(\mathbf{x}_1),\phi(\mathbf{x}_1)\rangle & \langle\phi(\mathbf{x}_1),\phi(\mathbf{x}_2)\rangle & \cdots & \langle\phi(\mathbf{x}_1),\phi(\mathbf{x}_n)\rangle \\ \langle\phi(\mathbf{x}_2),\phi(\mathbf{x}_1)\rangle & \langle\phi(\mathbf{x}_2),\phi(\mathbf{x}_2)\rangle & \cdots & \langle\phi(\mathbf{x}_2),\phi(\mathbf{x}_n)\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle\phi(\mathbf{x}_n),\phi(\mathbf{x}_1)\rangle & \langle\phi(\mathbf{x}_n),\phi(\mathbf{x}_2)\rangle & \cdots & \langle\phi(\mathbf{x}_n),\phi(\mathbf{x}_n)\rangle \end{pmatrix}$$

donant una matriu de similituds entre els elements de S que dependrà del *kernel* emprat.

Aquesta matriu serà l'element d'entrada de qualsevol algoritme d'aprenentatge *kernelitzat*, presentant tota la informació necessària per poder aplicar aquests mètodes.

Per tant, el càlcul de la matriu *kernel* serà el primer pas a l'hora d'aplicar un mè-

tode *kernel*. Per facilitar aquesta tasca podem utilitzar la funció `kernelMatrix` (`kernlab` [13]) implementada en R [15]. La funció `kernelMatrix` ens permet, donat un conjunt de dades i una funció *kernel* amb els paràmetres definits, obtenir la matriu *kernel* corresponent. Per a poder obtenir la matriu *kernel* resultant de la combinació de més d'un conjunt de dades s'ha implementat una versió ampliada de la funció `kernelMatrix` (Algoritme 1).

Algoritme 1: Matriu *kernel*

Donat l'element $D = \{S_1, \ldots, S_l\}$
amb $S_i = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \quad \forall\ i \in \{1, \ldots, l\}$, $\mathbf{x}_j \in \mathcal{X} \quad \forall\ j \in \{1, \ldots, n\}$ i $\mathcal{X} \subseteq \mathbb{R}^p$
Per cada conjunt de dades S_i de D
  definim el *kernel* a utilitzar: gaussià, polinòmic o ANOVA
  definim els paràmetres del kernel
    si el *kernel* és gaussià definim σ
    si el *kernel* és polinòmic definim α, d i c
    si el *kernel* és ANOVA definim σ i d
  calculem la matriu *kernel* ($\mathbf{K}_i$) amb la funció `kernelMatrix` (`kernlab`)
Si $l \geq 2$
  per cada conjunt de dades S_i de D
    donem un pes w_i a S_i
  construïm un nou kernel $\sum_{i=1}^{l} w_i \cdot \mathbf{K}_i$

Aquesta funció permetrà obtenir, a part de la matriu *kernel* d'un conjunt de dades, la matriu *kernel* de combinacions com

$$k(\mathbf{x}, \mathbf{x}') = \begin{cases} k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}') \\ w_1 k_1(\mathbf{x}, \mathbf{x}') + w_2 k_2(\mathbf{x}, \mathbf{x}') \end{cases}$$

si k_1 i k_2 són *kernels* en $\mathcal{X} \times \mathcal{X}$, amb $\mathcal{X} \subseteq \mathbb{R}^p$ i w_1 i w_2 són números reals positiu.

Així, un cop generada la matriu *kernel*, podrem utilitzar-la com a element d'entrada de mètodes de detecció de patrons *kernelitzats* com són les màquines de vectors de suport (SVM), l'anàlisi de components principals de *kernels* (KPCA) o l'anàlisi de correlació canònica de *kernels* (KCCA).

2.3. *Kernels* en aprenentatge no supervisat: *Kernel* PCA

L'aprenentatge no supervisat, com s'ha esmentat anteriorment, es caracteritza pel fet que els elements de la mostra d'aprenentatge no tenen una etiqueta que els classifiqui. Dins l'aprenentatge no supervisat trobem els mètodes de *clusteritzaci*ó, que busquen establir una tipologia en el conjunt dels elements de la mostra de manera que els grups resultants siguin homogenis internament i molt diferent entre ells; i les tècniques de reducció de dimensió, com l'anàlisi de components principals (PCA), que representen les observacions d'una mostra de grans dimensions en un espai reduït (2 o 3 dimensions) intentant conservar la disposició espacial de l'espai original. L'aplicació de *kernels* en aquests mètodes permetrà l'anàlisi de tot tipus de dades (vectors, textos, imatges, etc.) i l'exploració de relacions no lineals. Centrant-nos en l'anàlisi de components principals, parlarem de ***kernel* PCA (KPCA)** quan utilitzem les funcions *kernel* com a dades d'entrada d'aquest.

2.3.1 Anàlisi de components principals (PCA)

L'anàlisi de components principals (PCA) és un mètode de reducció de la dimensionalitat que busca reduir la dimensió de l'espai original generant un subespai en el que es trobin representades les dades originals amb la menor distorsió possible. A partir de les variables originals p, es crean unes noves variables r ($r < p$), anomenades components principals (PC), que són combinacions lineals de les variables originals, incorrelacionades entre elles, i que conserven la màxima variabilitat de les dades.

Suposant que disposem d'una mostra d'aprenentatge no etiquetada

$$S = \{\mathbf{x}_1, \dots \mathbf{x}_n\} \quad amb \quad \mathbf{x}_i \in \mathcal{X} \quad \forall \ i \in \{1, \dots n\} \quad i \quad \mathcal{X} \subseteq \mathbb{R}^{\mathrm{p}}$$

i que aquesta ha estat centrada

$$\sum_{i=1}^{n} \mathbf{x}_i = 0$$

La matriu de covariàncies ($\mathbf{C}$), de dimensió $p \times p$ vindrà donada per

$$\mathbf{C} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^T = \frac{1}{n} \mathbf{X}^T \mathbf{X}$$

A partir de les dades centrades i la matriu de covariàncies, es pot calcular la primera component principal (PC1), que es defineix com la combinació lineal de les variables originals que tenen màxima variància. Els valors d'aquesta primera component pels n individus es representaran per un vector $\mathbf{z}_1$

$$\mathbf{z}_1 = \mathbf{X}\mathbf{v}_1$$

Donat que les variables originals tenen mitjana zero, també $\mathbf{z}_1$ tindrà mitjana zero. La seva variància serà

$$\frac{1}{n}\mathbf{z}_1^T\mathbf{z}_1 = \frac{1}{n}\mathbf{v}_1^T\mathbf{X}^T\mathbf{X}\mathbf{v}_1 = \mathbf{v}_1^T\mathbf{C}\mathbf{v}_1$$

on $\mathbf{C}$ és la matriu de variàncies i covariàncies de les observacions.
En aquest punt, és obvi que es pot maximitzar la variància augmentant el mòdul del vector $\mathbf{v}_1$, però perquè la maximització tingui solució s'ha de definir el mòdul del vector $\mathbf{v}_1$ com $\mathbf{v}_1^T\mathbf{v}_1 = 1$.

Utilitzant el multiplicador de Lagrange

$$\mathbf{M} = \mathbf{v}_1^T\mathbf{C}\mathbf{v}_1 - \lambda\left(\mathbf{v}_1^T\mathbf{v}_1 - 1\right)$$

i maximitzant l'expressió tot derivant respecte els components de $\mathbf{v}_1$ i igualant a 0, tindrem

$$\frac{\partial \mathbf{M}}{\partial \mathbf{v}_1} = 2\mathbf{C}\mathbf{v}_1 - 2\lambda\mathbf{v}_1 = 0$$

amb solució

$$\mathbf{C}\mathbf{v}_1 = \lambda\mathbf{v}_1$$

on $\mathbf{v}_1$ és un vector propi de la matriu $\mathbf{C}$ i λ el valor propi corresponent.

Multiplicant per $\mathbf{v}_1^T$ ambdós costats s'obté

$$\mathbf{v}_1^T\mathbf{C}\mathbf{v}_1 = \lambda\mathbf{v}_1^T\mathbf{v}_1 = \lambda$$

pel que concloem que λ és la variància de $\mathbf{z}_1$. Com que aquesta és la quantitat que es vol maximitzar, λ serà el major valor propi de la matriu $\mathbf{C}$. El seu vector associat, $\mathbf{v}_1$, defineix els coeficients de cada variable en la primera component principal.

Així, a partir dels valors propis λ que obtindrem resolent el polinomi

$$p(\lambda) = |\mathbf{C} - \lambda\mathbf{I}| = 0$$

on $\mathbf{I}$ és la matriu identitat, pel que

$$\lambda\mathbf{I} = \Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_p \end{bmatrix} = \begin{bmatrix} Var(\mathbf{z}_1) & 0 & \cdots & 0 \\ 0 & Var(\mathbf{z}_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Var(\mathbf{z}_p) \end{bmatrix}$$

podem determinar els valors de λ i identificar el més gran, que correspondrà al valor propi de la component amb major variància.

Per trobar la segona component principal (PC2), definida com

$$\mathbf{z}_2 = \mathbf{X}\mathbf{v}_2$$

es podrà determinar igual que abans, agafant el següent valor de λ més gran. Com que aquesta serà la segona component, a més de definir que $\mathbf{v}_2^T\mathbf{C}\mathbf{v}_2$ sigui màxim i que $\mathbf{v}_2^T\mathbf{v}_2 = 1$, es demana que $\mathbf{v}_2$ sigui ortogonal a $\mathbf{v}_1$ perquè la segona component estigui intercorrelacionada amb la primera $\langle \mathbf{v}_2, \mathbf{v}_1 \rangle = 0$ (Figura 5).

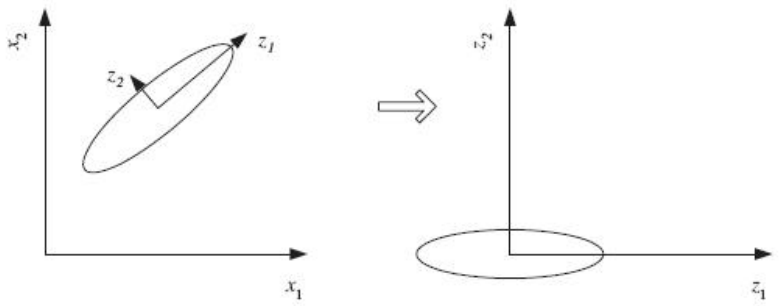

Figura 5: Anàlisi de components principals. Representació de les dues primeres components principals $(\mathbf{z}_1, \mathbf{z}_2)$. Les mostres es centren i seguidament es giren els eixos per alinear-los amb les direccions de major variancia [16] .

La resta de components, es trobaran de forma anàloga a la segona component (PC2), fins a un màxim de p components (sempre que $p > n$) [17].

Tots aquests càlculs per obtenir les components principals es poden realitzar automàticament utilitzant els algoritmes ja implementats en R. Trobem vàries llibreries de R que permeten fer un anàlisi de components principals, com: `stats` (funcions `prcomp` i `princomp`), `FactoMineR` (funció `PCA` [18]), `ade4` (funció `dudi.pca` [19]) i `amap` (funció `acp` [20]).

2.3.2 *Kernelització* del PCA

En el cas de l'anàlisi de components principals *kernelitzat* (*kernel* PCA), la idea és la mateixa que en el PCA, però en comptes d'utilitzar la matriu de covariàncies $\mathbf{C}$ per detectar el subespai de màxima variabilitat i mínima dimensió, s'utilitza la matriu dels productes escalars $\mathbf{K}$ (matriu *kernel*) que haurem obtingut d'una funció *kernel*.

Es demostra que

- és possible determinar els valors propis de la matriu de productes escalars ($\mathbf{K}$) centrada i, en conseqüència, la dispersió de les projeccions dels individus sobre les direccions de màxima variabilitat de l'espai vectorial $\mathbb{R}^p$ utilitzant tan sols la informació recollida en $\mathbf{K}$.
- es poden calcular les projeccions dels individus en aquestes direccions de màxima variabilitat a partir de la informació recollida en la matriu de productes escalars ($\mathbf{K}$) centrada.

fent possible l'ús de les funcions *kernel* en l'anàlisi de components principals [21–23].

Donat un conjunt de dades

$$S = \{\mathbf{x}_1, \dots \mathbf{x}_n\} \quad amb \quad \mathbf{x}_i \in \mathcal{X} \quad \forall\ i \in \{1, \dots n\} \quad i \quad \mathcal{X} \subseteq \mathbb{R}^p$$

en format matricial

$$\mathbf{X} = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{1n} & \cdots & x_{np} \end{bmatrix}$$

i que s'ha definit una funció ϕ (*embedding function*)

$$\begin{array}{rccc} \phi: & \mathcal{X} & \rightarrow & \mathcal{F} \\ & \mathbf{x} & \rightarrow & \phi(\mathbf{x}) \end{array}$$

que projecta els elements de S en un espai $\mathcal{F}$, tot generant un nou conjunt de dades

$$\phi\left(S\right) = \left\{\phi\left(\mathbf{x}_1\right), \ldots, \phi\left(\mathbf{x}_n\right)\right\} \quad amb \quad \phi(\mathbf{x}_i) \in \mathcal{F} \quad \forall\ i \in \{1, \ldots, n\}$$

Si suposem que aquestes dades estan centrades en $\mathcal{F}$

$$\tilde{\phi}(\mathbf{x}_i) = \phi(\mathbf{x}_i) - \bar{\phi}$$

$$\sum_{i=1}^{n} \tilde{\phi}(\mathbf{x}_i) = 0$$

La matriu de covariàncies en $\mathcal{F}$ es defineix com

$$\mathbf{C}_F = \frac{1}{n} \sum_{i=1}^{n} \tilde{\phi}(\mathbf{x}_i) \tilde{\phi}(\mathbf{x}_i)^T$$

Per trobar els valors propis no nuls ($\lambda > 0$) i els vectors propis $\mathbf{v}$ respectius haurem de resoldre

$$\lambda \mathbf{v} = \mathbf{C}_F \mathbf{v}$$

Definint el vector propi com una combinació lineal de característiques

$$\mathbf{v} = \sum_{i=1}^{n} \alpha_i \tilde{\phi}(\mathbf{x}_i)$$

i multiplicant als dos costats per $\tilde{\phi}(\mathbf{x}_k)$ transposat, obtindrem

$$\lambda \sum_{i=1}^{n} \alpha_i \langle \tilde{\phi}(\mathbf{x}_k), \tilde{\phi}(\mathbf{x}_i) \rangle =$$

$$\frac{1}{n} \sum_{i=1}^{n} \alpha_i \langle \tilde{\phi}(\mathbf{x}_k), \sum_{j=1}^{n} \tilde{\phi}(\mathbf{x}_j) \rangle \langle \tilde{\phi}(\mathbf{x}_j), \tilde{\phi}(\mathbf{x}_i) \rangle, \qquad \forall\ k = \{1, \ldots, n\}$$

Definint la matriu $\tilde{\mathbf{K}}$ com $\tilde{k}_{ij} = \langle \tilde{\phi}(\mathbf{x}_j), \tilde{\phi}(\mathbf{x}_i) \rangle$, s'obté

$$n\lambda \tilde{\mathbf{K}} \boldsymbol{\alpha} = \tilde{\mathbf{K}}^2 \boldsymbol{\alpha}$$

Degut a la simetria de $\tilde{\mathbf{K}}$, els seus vectors propis generen l'espai complet, pel que

$$n\lambda \boldsymbol{\alpha} = \tilde{\mathbf{K}} \boldsymbol{\alpha}$$

on els valors propis associats a $\boldsymbol{\alpha}$ corresponen a $n\lambda$, pel que serà necessari aplicar la restricció $\| \mathbf{v} \| = 1$ als vectors propis de $\tilde{\mathbf{K}}$

$$1 = \sum_{i,j=1}^{n} \alpha_i \alpha_j \langle \tilde{\phi}(\mathbf{x}_i), \tilde{\phi}(\mathbf{x}_j) \rangle = \lambda \langle \boldsymbol{\alpha}, \boldsymbol{\alpha} \rangle$$

de manera que obtindrem les components principals del *kernel*, utilitzant

$$\langle \mathbf{v}_k, \tilde{\phi}(\mathbf{x}) \rangle = \sum_{i=1}^{n} \alpha_{ik} \langle \tilde{\phi}(\mathbf{x}_i), \tilde{\phi}(\mathbf{x}) \rangle$$

on $k = 1, \ldots, r$; essent els vectors propis.

Com que les dades centrades $\tilde{\phi}(\mathbf{x}_i)$ no es coneixen, generalment ϕ és desconegut, no podrem calcular la matriu $\tilde{\mathbf{K}}$ de forma explícita. No obstant, podrem solventar-ho a partir de la seva homòloga no centrada $\mathbf{K}$.

$$\begin{aligned}
\tilde{k}_{ij} &= \langle \tilde{\phi}(\mathbf{x}_i), \tilde{\phi}(\mathbf{x}_j) \rangle \\
&= \langle \phi(\mathbf{x}_i) - \bar{\phi}, \phi(\mathbf{x}_j) - \bar{\phi} \rangle \\
&= \left(\phi(\mathbf{x}_i) - \tfrac{1}{n} \textstyle\sum_{i=1}^{n} \phi(\mathbf{x}_i) \right) \left(\phi(\mathbf{x}_i) - \tfrac{1}{n} \textstyle\sum_{i=1}^{n} \phi(\mathbf{x}_i) \right)^T \\
&= k_{ij} - \tfrac{1}{n} \textstyle\sum_{l=1}^{n} k_{il} - \tfrac{1}{n} \textstyle\sum_{l=1}^{n} k_{jl} + \tfrac{1}{n^2} \textstyle\sum_{l,t=1}^{n} k_{lt}
\end{aligned}$$

Que podem escriure de forma més compacta utilitzant el vector $\mathbf{1}_n = (1, \ldots, 1)^T$

$$\tilde{\mathbf{K}} = \mathbf{K} - \frac{1}{n} \mathbf{K} \mathbf{1}_n \mathbf{1}_n^T - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T \mathbf{K} + \frac{1}{n^2} \left(\mathbf{1}_n^T \mathbf{K} \mathbf{1}_n \right) \mathbf{1}_n \mathbf{1}_n^T$$

Obtinguts els components principals, en el cas del *kernel* PCA haurem de determinar la projecció de $\tilde{\phi}(\mathbf{x})$. Donat l'element $\mathbf{y}$ i sent $\mathbf{z} = \left(\langle \tilde{\phi}(\mathbf{x}_i), \tilde{\phi}(\mathbf{y}) \rangle\right)_{n\times 1}$ i $\mathbf{V}$ la matriu $n \times r$ que conté els vectors propis $\mathbf{v}_l$, la projecció en les r components s'expressa com

$$\left(\langle \mathbf{v}_l, \tilde{\phi}(\mathbf{y}) \rangle\right)_{1\times r} = \left(\mathbf{z}^T - \frac{1}{n}\mathbf{1}_n^T\mathbf{K}\right)\left(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\mathbf{V}$$

Així, el *kernel* PCA ens permetrà fer l'extensió no lineal del PCA (Figura 6).

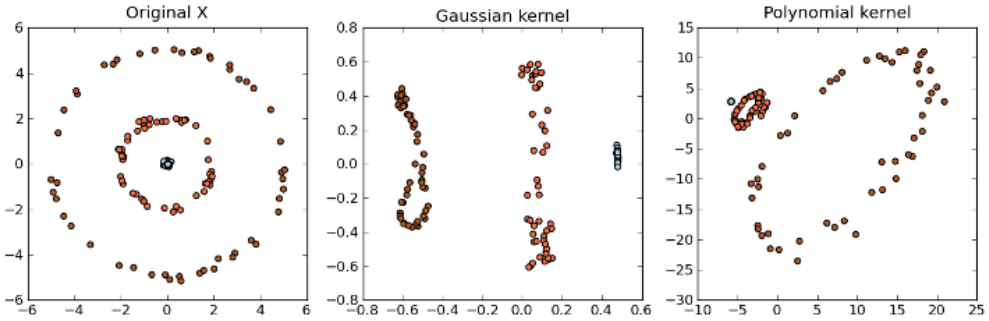

Figura 6: Representació gràfica d'un anàlisi de components principals amb *kernels* (KPCA). A l'esquerra es mostren les dades originals (linealment no separables). Al centre el PCA resultant d'aplicar un *kernel* gaussià a les dades d'origen, aconseguint una separació lineal de les dades transformades. A la dreta es mostra el resultat d'aplicar un *kernel* diferent a l'anterior, el *kernel* polinòmic, generant una nova distribució espacial de les mostres.

Trobem una llibreria en R, anomenada `kernlab` [13], que mitjançant la funció `kpca` ens permet obtenir tota la informació relacionada amb l'anàlisi de components principals amb *kernels*. A partir d'una matriu *kernel*, aquesta funció ens proporcionarà la matriu amb les components principals (argument `pcv`), els valors propis corresponents a cada component principal (argument `eig`) i les coordenades dels punts de les components principals (argument `rotated`). La funció també accepta com a informació d'entrada el conjunt de dades original i la funció *kernel* amb els paràmetres definits.

A més, utilitzant la matriu *kernel* resultant de l'Algoritme 1, podrem fer l'anàlisi de components principals de la matriu *kernel* obtinguda de la combinació de més d'un *kernel*.

Tot i els avantatges que ens proporciona el *kernel* PCA per la versatilitat i les propietats associades a les funcions *kernel*, a la pràctica ens trobem amb un parell d'inconvenients que cal remarcar. Per una banda, la manca d'interpretabilitat de les variables originals, ja que les dades han estat transformades per la funció *kernel* i estan representades en un nou espai $\mathcal{F}$. I per altra banda, la necessitat de determinar el *kernel* i els paràmetres òptims per a les dades en estudi.

3. Descobrint variables

Un estudi publicat recentment presenta una nova estratègia per abordar la manca d'interpretabilitat del *kernel* PCA [6]. El mètode que es planteja permet la representació de les variables originals en el *biplot* resultant del *kernel* PCA. A més a més, també es proporciona una estratègia pel descobriment de noves variables d'interès. Ambdós mètodes es descriu a continuació.

3.1. Representació de variables originals

Per a millorar la interpretabilitat del resultat del *kernel* PCA, es proposa un procediment que permet representar per cada variable la direcció de màxim creixement a nivell local per cada un dels individus en estudi.
Considerant que les observacions són realitzacions del vector aleatori $\mathbf{X} = (X_1, \ldots, X_p)$, es vol representar la prominència de la variable X_k en el *kernel* PCA.

Es defineix un conjunt de punts de la forma

$$\mathbf{y} = \mathbf{a} + s\mathbf{e}_k \in \mathbb{R}^p$$

on $\mathbf{e}_k = (0, \ldots, 1, \ldots) \in \mathbb{R}^p$, essent la $k-$èssima component igual a 1 i la resta 0, $s \in \mathbb{R}$ i $\mathbf{a} \in \mathbb{R}^p$.

Podrem representar les imatges d'aquest element $\tilde{\phi}(\mathbf{y})$, utilitzant

$$\left(\langle \mathbf{v}_l, \tilde{\phi}(\mathbf{y})\rangle\right)_{1\times r} = \left(\mathbf{z}_s^T - \frac{1}{n}\mathbf{1}_n^T\mathbf{K}\right)\left(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\mathbf{V}$$

on $\mathbf{z}_s = \left(\langle \tilde{\phi}(\mathbf{x}_i), \tilde{\phi}(\mathbf{y})\rangle\right)_{n\times 1}$

Aleshores, podrem representar la direcció de màxim creixement de $\sigma^k(s)$ per la variable X_k projectant el vector tangent quan $s = 0$.

$$\left.\frac{d\sigma^k}{ds}\right|_{s=0} = \left.\frac{d\mathbf{z}_s^T}{ds}\right|_{s=0}\left(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)\mathbf{V}$$

On

$$\left.\frac{d\mathbf{z}_s^T}{ds}\right|_{s=0} = \left(\left.\frac{dz_s^1}{ds}\right|_{s=0}, \ldots, \left.\frac{dz_s^n}{ds}\right|_{s=0}\right)^T$$

Utilitzant la regla de la cadena, s'obté

$$\left.\frac{dz_s^i}{ds}\right|_{s=0} = \left.\frac{\partial K(\mathbf{y}, \mathbf{x}_i)}{\partial y_k}\right|_{y=a}$$

on el resultat de la derivada dependrà del *kernel* utilitzat. Un cop resolta la derivada i substituint $\mathbf{a} = \mathbf{x}_\beta$ (punt d'entrenament), podrem projectar la direcció de màxim creixement de la variable.

El resultat particular de $\left.\frac{dz_s^i}{ds}\right|_{s=0}$ pel *kernel gaussià*, polinòmic i lineal serà:

- <u>*Kernel* gaussià</u>. Donada la funció del *kernel* gaussià

$$k(\mathbf{y}, \mathbf{x}_i) = exp\left(-\sigma||\mathbf{y} - \mathbf{x}_i||^2\right) = exp\left(-\sigma\sum_{j=1}^{p}(y_j - x_{ij})^2\right)$$

pel conjunt de punts de la forma $\mathbf{y} = \mathbf{a} + s\mathbf{e}_k \in \mathbb{R}^p$ s'obté

$$\left.\frac{dz_s^i}{ds}\right|_{s=0} = \left.\frac{\partial K(\mathbf{y}, \mathbf{x}_i)}{\partial y_k}\right|_{\mathbf{y}=\mathbf{a}} = -2\sigma k(\mathbf{a}, \mathbf{x}_i)(a_k - x_{ik})$$

i si $\mathbf{a} = \mathbf{x}_\beta$ (un punt d'entrenament), aleshores

$$\left.\frac{dz_s^i}{ds}\right|_{s=0} = -2\sigma k(\mathbf{x}_\beta, \mathbf{x}_i)(x_{\beta k} - x_{ik})$$

- *Kernel* polinòmic Donada la funció del *kernel* polinòmic

$$k(\mathbf{y}, \mathbf{x}_i) = (\alpha \cdot \langle \mathbf{y}, \mathbf{x}_i \rangle + c)^d$$

pel conjunt de punts de la forma $\mathbf{y} = \mathbf{a} + s\mathbf{e}_k \in \mathbb{R}^p$ s'obté

$$\left.\frac{dz_s^i}{ds}\right|_{s=0} = \left.\frac{\partial K(\mathbf{y}, \mathbf{x}_i)}{\partial y_k}\right|_{\mathbf{y}=\mathbf{a}} = d\left(\alpha \mathbf{x}_i^T \mathbf{a} + c\right)^{d-1} \alpha x_{ik}$$

i si $\mathbf{a} = \mathbf{x}_\beta$ (un punt d'entrenament), aleshores

$$\left.\frac{dz_s^i}{ds}\right|_{s=0} = d\left(\alpha \mathbf{x}_i^T \mathbf{x}_\beta + c\right)^{d-1} \alpha x_{ik}$$

- *Kernel* lineal Donada la funció del *kernel* lineal

$$k(\mathbf{y}, \mathbf{x}_i) = \langle \mathbf{y}, \mathbf{x}_i \rangle$$

pel conjunt de punts de la forma $\mathbf{y} = \mathbf{a} + s\mathbf{e}_k \in \mathbb{R}^p$ s'obté

$$\left.\frac{dz_s^i}{ds}\right|_{s=0} = \left.\frac{\partial K(\mathbf{y}, \mathbf{x}_i)}{\partial y_k}\right|_{\mathbf{y}=\mathbf{a}} = x_{ik}$$

D'aquesta manera es podran representar les variables en el *kernel* PCA.

Com a il·lustració d'aquesta metodologia es fa servir un conjunt de dades d'un estudi d'expressió gènica en ratolins on s'analitza l'efecte de 5 dietes lipídiques diferents en dos grups de ratolins (*wild-type* versus mutats (PPARα)). Els resultats mostrats a la Figura 7 permeten identificar variables que es comporten de forma similar entre les observacions, i particularment, que apunten cap a una agrupació d'observacions, indicant diferències en els valors d'aquesta variable entre grups.

La llargada de les fletxes (vectors) que representen les variables variarà en funció de la variable que es representi, ja que dependrà de la situació d'aquesta variable en l'espai de projecció. Si es vol veure una variable en concret, si és necessari, es podrà deformar l'espai tot modificant el *kernel* utilitzat per tal d'aconseguir una millor representació d'aquesta.

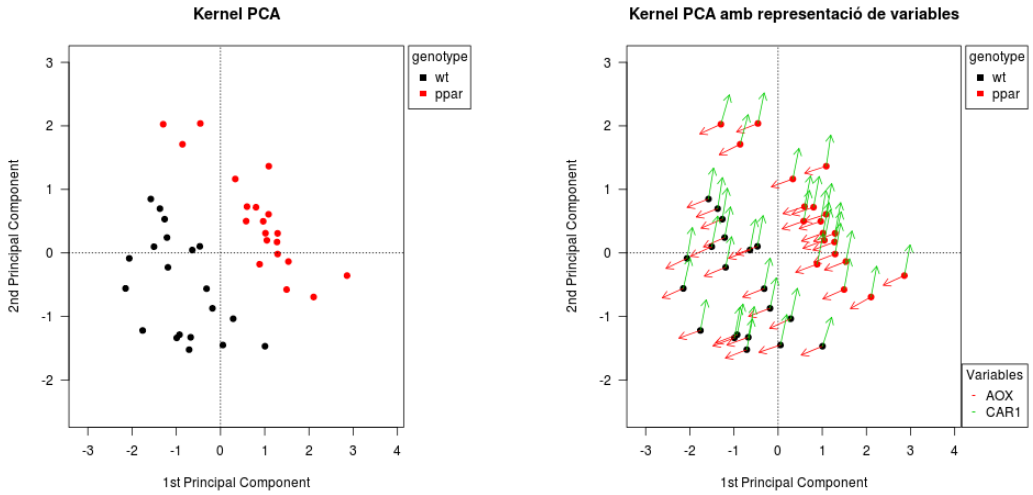

Figura 7: Representació d'un anàlisi *kernel* PCA amb representació de variables. A partir de les dades d'un estudi d'expressió gènica en ratolins on s'analitza l'efecte de 5 dietes lipídiques diferents en dos grups de ratolins (*wild-type* versus mutats (PPARα)), es mostra el resultat del *kernel* PCA utilitzant un *kernel* gaussià amb $\sigma = 0.05$ (gràfic de l'esquerra) i seguidament es mostra la representació de dues variables (2 gens): AOX i CAR1 (gràfic de la dreta); indicant que AOX s'expressa més en el grup *wild-type* i CAR1 en el grup de ratolins mutats [24].

Per tal d'obtenir la representació de variables en el *kernel* PCA s'ha implementat una funció en R per facilitar aquesta tasca (Algoritme 2).

Es pot fer una extensió de la representació de variables mitjançant la representació de **combinacions lineals de variables**.

Suposant que es vol representar la combinació lineal

$$X_{k_1} + X_{k_2} + \ldots + X_{k_l}$$

on $k_1, k_2, \ldots, k_l \in \{1, 2, \ldots, p\} \quad amb \quad k_i \neq k_j, \quad i, j = \{1, \ldots l\}$.

Pel *kernel* gaussià tindrem que

$$\left.\frac{d\mathbf{z}_s^i}{ds}\right|_{s=0} = \sum_{t=1}^{l} \left.\frac{\partial K(\mathbf{y}, \mathbf{x}_i)}{\partial y_{k_t}}\right|_{\mathbf{y}=\mathbf{a}}$$

Així, per aquest cas en particular, es podrà implementar fàcilment en l'Algoritme

2 tot fent un sumatori en

$$\left.\frac{\partial K(\mathbf{y}, \mathbf{x}_i)}{\partial y_k}\right|_{\mathbf{y}=\mathbf{a}}$$

Algoritme 2: Representació de variables

Donat l'objecte D resultat de la funció `kpca` (de la llibraria `kernlab`)
es determinen els punts d'entrenament ($\mathbf{x}_\beta$) que corresponen a les projeccions de $\tilde{\phi}(\mathbf{x})$

$$\mathbf{x}_\beta = \texttt{rotated}(D)$$

es determina la matriu $(k \times r)$ amb els components principals

$$\mathbf{V} = \texttt{pcv}(D)$$

es calcula la matriu $\mathbf{M}$ $(n \times n)$

$$\mathbf{M} = \left(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T\right)$$

es determina, particular a cada *kernel*,

$$\left.\frac{\partial K(\mathbf{y}, \mathbf{x}_i)}{\partial y_k}\right|_{\mathbf{y}=\mathbf{a}} \quad amb \quad \mathbf{a} = \mathbf{x}_\beta$$

calculem les projeccions de màxim creixement de la variable per cada individu

$$\left.\frac{d\sigma^k}{ds}\right|_{s=0} = \left.\frac{d\mathbf{z}_s^T}{ds}\right|_{s=0} \mathbf{MV}$$

proporcionem les coordenades de la direcció de màxim creixement

origen $= \mathbf{x}_\beta$

fi $= \mathbf{x}_\beta + \sigma(s)$

Què passarà si volem treballar amb més d'un conjunt de dades? Utilitzant l'Algoritme 1 haurem obtingut la matriu *kernel* corresponent a la integració dels conjunts de dades.

Aleshores, sempre que el *kernel* utilitzat per transformar les dades sigui el mateix, tot i que els paràmetres variïn, es podran representar les dades de cada un dels conjunts igual que quan es treballa amb un únic conjunt de dades. Simplement, es definirà

$$\left.\frac{d\mathbf{z}_s^i}{ds}\right|_{s=0} = \left.\frac{\partial K_l(\mathbf{y}_l, \mathbf{x}_{li})}{\partial y_{lk}}\right|_{\mathbf{y}_l=\mathbf{a}_l}$$

on l representa el conjunt de dades del que prové la variable que es vol representar, $l = 1, 2, \ldots$.

3.2. Cerca de variables d'interès

El procediment descrit permet representar el vector que mostra la direcció de màxim creixement d'una variable en un punt donat, concretament, el de cada observació.

Es pot pensar, per tant, que si definim una direcció d'interès en el pla, donada per un vector $\mathbf{w}$, podem obtenir aquelles variables representades en el *kernel* PCA que es correlacionin amb $\mathbf{w}$.

Si ens fixem en la Figura 7, on observem dues agrupacions de punts (vermells i negres) de manera que els individus queden estratificat en funció del seu genotip, podría ser interessant identificar aquelles variables associades a cada un d'aquests genotips. Per tant, definint un vector $\mathbf{w}$ entre els punts centrals (centroides) de cada clúster, podrem trobar aquelles variables que millor es correlacionen amb $\mathbf{w}$, essent variables altament implicades en la distribució espacial de les mostres (Figura 8).

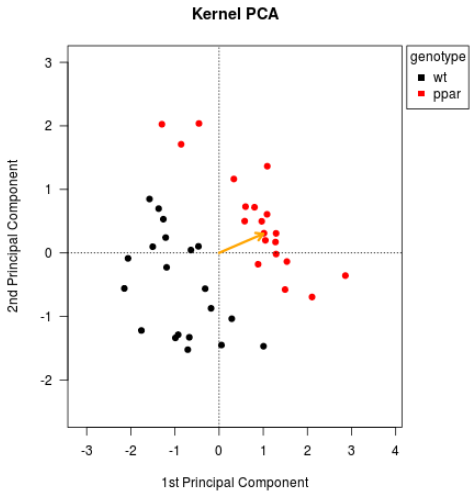

Figura 8: Representació d'un anàlisi *kernel* PCA amb descobriment de variables.

Donat un vector $\mathbf{w}$ en un *kernel* PCA, es defineix $\mathbf{w}_i$ com el vector paral·lel a $\mathbf{w}$ lligat a la imatge de la observació $\mathbf{x}_i, \quad i = \{1, \ldots, n\}$. Aleshores, per cada variable X_k podem calcular el vector que representa la direcció de màxim creixement d'aquesta variable per a cada individu segons el descrit anteriorment

$$\left.\frac{d\sigma^k}{ds}\right|_{s=0}$$

pel que obtindrem 2 vectors per cada mostra. Un corresponent a $\mathbf{w}_i$ i l'altre a la variable X_k, $\left(\left.\frac{d\sigma^k}{ds}\right|_{s=0}\right)_{x_i}$.

Podrem mesurar la correlació entre X_k i $\mathbf{w}$ a partir del cosinus obtingut per cada parell de vectors. Finalment, es podrà obtenir un valor de correlació per a cada variable a partir de la mitjana dels cosinus de les observacions.

$$R_k := \frac{1}{n}\sum_{i=1}^{n} cos\left(\mathbf{w}_i, \left(\left.\frac{d\sigma^k}{ds}\right|_{s=0}\right)_{x_i}\right)$$

La metodologia proposada per a la cerca de variables a partir d'una direcció d'interès definida en el pla s'ha implementat en R per facilitar-ne el càlcul (Algoritme 3).

Algoritme 3: Cerca de variables d'interès

Donat un vector $\mathbf{w}$ d'interès

Definir $\mathbf{w}_i$ per cada observació $\mathbf{x}_i$

Donada la matriu *kernel* resultant de l'Algoritme 1

per cada variable

per cada individu

calculem la projecció de màxim creixement per cada individu (Algoritme 2)

$$\left.\frac{d\sigma^k}{ds}\right|_{s=0} = \left.\frac{d\mathbf{z}_s^T}{ds}\right|_{s=0} \mathbf{MV}$$

calculem el cosinus entre els 2 vectors

$$cos\left(\mathbf{w}_i, \left(\left.\frac{d\sigma^k}{ds}\right|_{s=0}\right)_{x_i}\right) = \frac{\sum_{i=1}^n x_i \cdot y_i}{\sqrt{\sum_{i=1}^n x_i^2} \cdot \sqrt{\sum_{i=1}^n y_i^2}}$$

obtenim la mitja dels alineaments per cada variable

$$R_k := \frac{1}{n}\sum_{i=1}^{n} cos\left(\mathbf{w}_i, \left(\left.\frac{d\sigma^k}{ds}\right|_{s=0}\right)_{x_i}\right)$$

Capítol 2. Software estadístic

1. Software estadístic R

El R és un llenguatge orientat al processament i anàlisi estadístic de dades. Es caracteritza per ser desenvolupat de forma lliure i gratuïta i publicar-se sota llicència GNU GPL. El projecte va començar fa més de vint anys i hi participen membres de varis països. Actualment, ofereix un total de 6.657 llibreries proporcionant als usuaris diverses eines d'alta qualitat per a l'anàlisi estadístic i la representació gràfica de les dades.

El 2001 neix un projecte lligat a R, anomenat Bioconductor, que sota la mateixa filosofia agrupa codi obert per a l'anàlisi i comprensió de dades òmiques. Actualment, Bioconductor proporciona 1.024 llibreries englobant un ampli ventall de potents mètodes estadístics i gràfics per a l'anàlisi de dades de microarrays, citometria de flux i *real-time* RT-PCR, entre altres.

La transparència i el lliure ús d'aquest llenguatge afavoreix una constant millora i desenvolupament de les eines que ofereix, essent possible gràcies a la interacció entre els diferents usuaris i membres del projecte. Amb l'objectiu de divulgar les eines descrites en el Capítol 1 per a millorar la interpretabilitat del *kernel* PCA i proporcionar eines per a la integració de dades basades en *kernels*, s'ha creat la corresponent llibreria en R. La llibreria creada s'anomena `KPCAplus` i conté la funció en R (`KPCAplus`) que permet fer els anàlisis esmentats dins d'aquest entorn. A més a més, facilita una aplicació web d'aquesta (`KPCAplusGUI`) amb la corresponent documentació en HTML per a fer l'anàlisi més interactiu i assequible a un major nombre d'usuaris.

2. Funció `KPCAplus`

`KPCAplus` és la funció resultant d'implementar en R els algoritmes que s'han descrit anteriorment per a:

- integrar conjunts de dades mitjançant *kernels*
- representar les variables originals en el *kernel* PCA
- cercar variables d'interès

amb l'objectiu de facilitar-ne i extendre'n l'ús.

A continuació es descriu la funció i els seus arguments. El codi de la funció `KPCAplus` es pot veure a l'Annex.

```
KPCAplus(x, kernel_function, kparameters, xpc = 1, ypc = 2,
factors = NULL, variables = NULL, combination = "+",
var_combination = NULL, coef_combination = NULL, scale = 0.5,
plot.kPCA = TRUE, plot.profile = FALSE, variables_discovery = NULL,
discovery.col = "orange", weights = NULL, text.sample = FALSE,
show.legend = TRUE, ...)
```

Amb els arguments:

`x`
: matriu de dades indexada per files. En cas de voler obtenir el *kernel* PCA de més d'un conjunt de dades es donarà una llista de matrius (cada objecte de la llista és una matriu de dades).

`kernel_function`
: funció *kernel* a utilitzar entre

  - `rbfdot`, *kernel* gaussià
  - `polydot`, *kernel* polinòmic
  - `anovadot`, *kernel* ANOVA

  En cas de tenir més d'un conjunt de dades es donarà una llista amb el *kernel* de cada conjunt de dades. El *kernel* definit en el primer argument de la

llista s'aplicarà en la matriu de dades que es trobi en el primer argument de la llista `x`, el segon argument de la llista s'aplicarà en la matriu de dades que es trobi en el segon argument de la llista `x`, i així successivament.

`kparameters`

llista amb els paràmetres corresponents al *kernel* definit a `kernel_function`. Els paràmetres a definir per cada funció *kernel* són:

- `sigma` pel *kernel* gaussià (`rbfdot`)
- `degree`, `scale` i `offset` pel *kernel* polinòmic (`polydot`)
- `sigma` i `degree` pel *kernel* ANOVA (`anovadot`)

En cas de tenir més d'un conjunt de dades es donarà una llista de llistes de paràmetres *kernel*. La llista de paràmetres definits en el primer argument de la llista correspondrà a la funció *kernel* que es trobi en el primer argument de la llista `kernel_function`, la llista de paràmetres definits en el segon argument de la llista correspondrà a la funció *kernel* que es trobi en el segon argument de la llista `kernel_function`, i així successivament.

`xpc`

número indicant la component principal a representar en l'eix de les abscisses, prenent com a valor màxim el 8 (PC8). Per defecte: `1`.

`ypc`

número indicant la component principal a representar en l'eix de les ordenades, prenent com a valor màxim el 8 (PC8). Per defecte: `2`.

`factors`

data frame ($n \times k$), on n correspon al nombre d'observacions del conjunt de dades i k són les variables categòriques associades al conjunt de dades. Si $k > 2$, només les dues primeres variables s'utilitzaran com a factors en la representació gràfica del *kernel* PCA. Per defecte: `NULL`.

`variables`

nom de les variables del/s conjunt/s de dades analitzades que es volen representar en el *biplot* del *kernel* PCA. Per defecte: `NULL`.

`combination`
: tipus de combinació lineal de variables que es vol realitzar: sumatori (`+`), resta (`-`), multiplicació (`*`). Per defecte: `+`.

`var_combination`
: nom de les variables del/s conjunt/s de dades analitzades que es volen representar com a combinació lineal en el *biplot*. Per defecte: `NULL`.

`coef_combination`
: vector numèric amb els coeficients de les variables quan estem representant una combinació lineal de variables. Per defecte: `NULL`.

`scale`
: valor numèric definint l'escalat de la longitud de les fletxes representades en el *kernel* PCA. Per defecte: `0.5`

`plot.kPCA`
: valor lògic especificant la visualització del *biplot*. Per defecte: `TRUE`.

`plot.profile`
: valor lògic especificant la visualització del perfil de les variables definides en l'argument `variables`. Si `variables=NULL` aquesta representació no és viable. Per defecte: `FALSE`.

`variables_discovery`
: vector numèric indicant les coordenades d'un vector numèric, de la forma $\mathbf{v} = (x_0, y_0, x_1, y_1)$, corresponents a una direcció d'interès en el pla. Per defecte: `NULL`.

`discovery.col`
: color del vector corresponent a `variables_discovery`.
Per defecte: `orange`.

`weights`
: vector numèric definit quan es té més d'un conjunt de dades. Permet donar pesos diferents a cada conjunt de dades. Per defecte: `NULL`.

`text.sample`
: valor lògic especificant la visualització del nom de les observacions (`rownames(x)`) en el *biplot* resultant del *kernel* PCA. Per defecte: `FALSE`.

`show.legend`
: valor lògic especificant la visualització de la llegenda en el *biplot*. Per defecte: `TRUE`.

`...`
: paràmetres adicionals de la funció *kpca* (*kernlab*).

Així, especificant els diferents arguments de la funció podrem obtenir el *kernel* PCA d'una combinació de dades, la representació de variables en un *kernel* PCA, la representació de combinacions lineals de variables i finalment, descobrir variables d'interès definint un vector en el pla del *kernel* PCA obtingut.

Els valors que obtindrem de la funció `KPCAplus` se'ns retornaran en una llista que contindrà:

`datasets`
: recull la informació de les dades analitzades, incloent la matriu de dades (`x`) i el nombre de conjunts de dades utilitzats.

`KPCA`
: conté tota la informació relacionada amb el *kernel* PCA: la funció *kernel*, els paràmetres del *kernel*, la matriu *kernel*, els valors propis, components principals i les projeccions del *kernel* PCA. En cas d'analitzar més d'un conjunt de dades ens indicarà, a més a més, els pesos (`weights`) utilitzats.

`Variable discovery`
: conté els resultats referents a la cerca de noves variables. Recull la informació en format `data.frame` de 4 columnes i p files (variables). A les diferents columnes trobarem el nom de les variables, un número referent al conjunt de dades al que pertany cada variable (en cas de fer una anàlisi integrant varis conjunts de dades), el valor mitjà de les correlacions i la desviació estàndard.

3. Llibreria `shiny`

Amb l'objectiu de fer menys tediós l'anàlisi de dades en R, recentment s'han desenvolupat llibreries que proporcionen eines per a la creació d'interfícies gràfiques (GUI) per a les funcions en R, com `gWidgets` i `shiny`.

La llibreria `shiny` [25], desenvolupada per RStudio [26], permet crear aplicacions web interactives en R de forma molt simple i amb un resultat estètic bo. Així, per poder obtenir l'aplicació web d'un programa en R, només es necessita construir dos fitxers:

`ui.R`
: defineix la interfície gràfica de l'aplicació.

`server.R`
: conté el codi R corresponent a la funció desenvolupada, tot definint les entrades i sortides de l'algoritme.

Amb la intenció d'apropar l'exploració de dades a través del *kernel* PCA a tot tipus d'usuaris, tant els que tenen coneixements de R com els que no, s'ha implementat l'aplicació web utilitzant aquesta llibreria.

La interfície web que s'ha dissenyat permet realitzar el *kernel* PCA per a un conjunt de dades pròpies de l'usuari (màxim 3 conjunts de dades) o executar un parell d'exemples en els que s'analitza un conjunt de dades o dos. Visualment, l'aplicació s'ha estructurat en dues parts, el lateral esquerra on s'agrupen els diferents paràmetres que ha de definir l'usuari per a l'anàlisi, i l'espai restant, que mostra els resultats del *kernel* PCA en diferents pestanyes (Figura 9).

Aquestes són:

Data files
: Mostra una descripció del conjunt de dades que s'està analitzant permetent, en cas de treballar amb un conjunt de dades propi, comprovar que aquest s'estigui carregant correctament. Dóna el nom del/s fitxer/s a analitzar i la dimensió de cada un d'aquests (nombre d'observacions $\times$ nombre de variables), i mostra les primeres files i columnes de la matriu de dades (Figura 9).

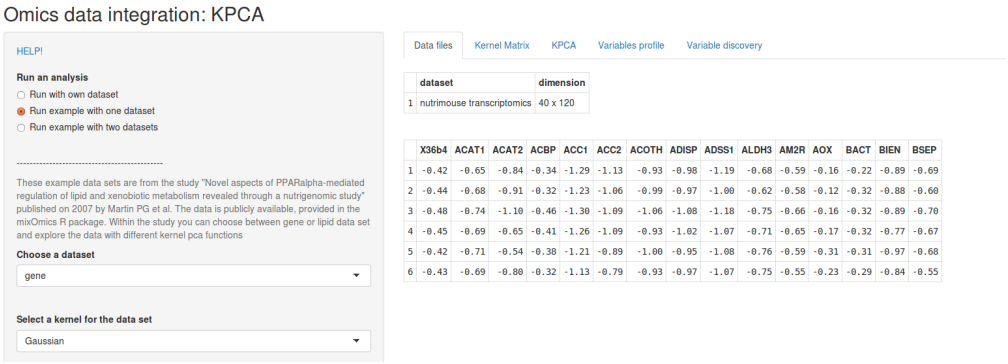

Figura 9: Interfície gràfica de l'aplicació. Al lateral esquerra s'observen els diferents paràmetres que ha de definir l'usuari i a l'espai restant les diferents pestanyes que mostraran els resultats de l'anàlisi, en aquest cas, la descriptiva del conjunt de dades.

Kernel Matrix

Descriptiu de la matriu *kernel* obtinguda. Aquesta pestanya ens mostra els valors descriptius bàsics (mínim, màxim, IQR, mitjana i mediana), un histograma i una representació en un rang de colors dels valors de la matriu *kernel* (Figura 10). Tota aquesta informació ens ajudarà a avaluar la funció *kernel* i els paràmetres a aplicar a les dades.

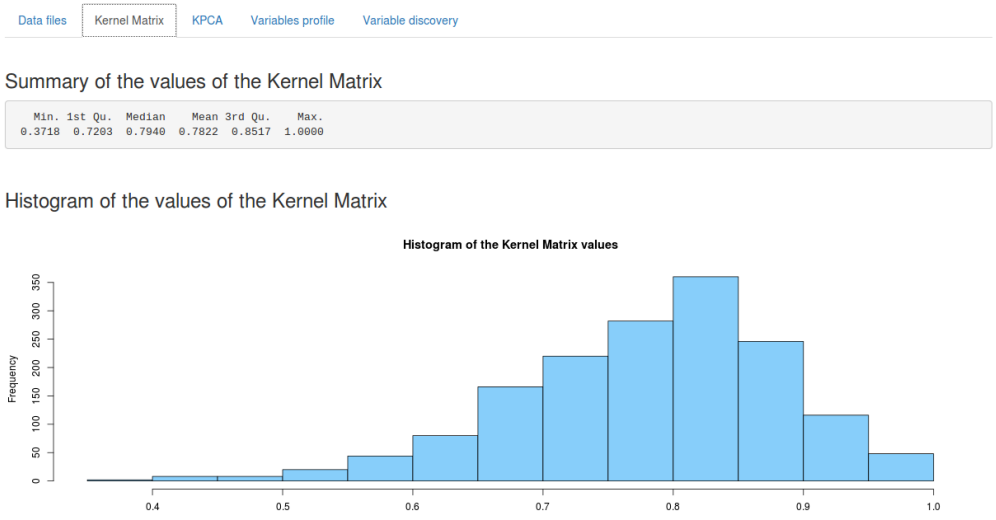

Figura 10: Descriptiu de la matriu *kernel* i histograma.

KPCA

Representació en 2D (*biplot*) de les projeccions de les dues primeres components principals del *kernel* PCA. Interactivament permetrà visualitzar la representació de variables, combinacions lineals de variables i, en cas d'estar definit, el vector pel descobriment de variables (Figura 11).

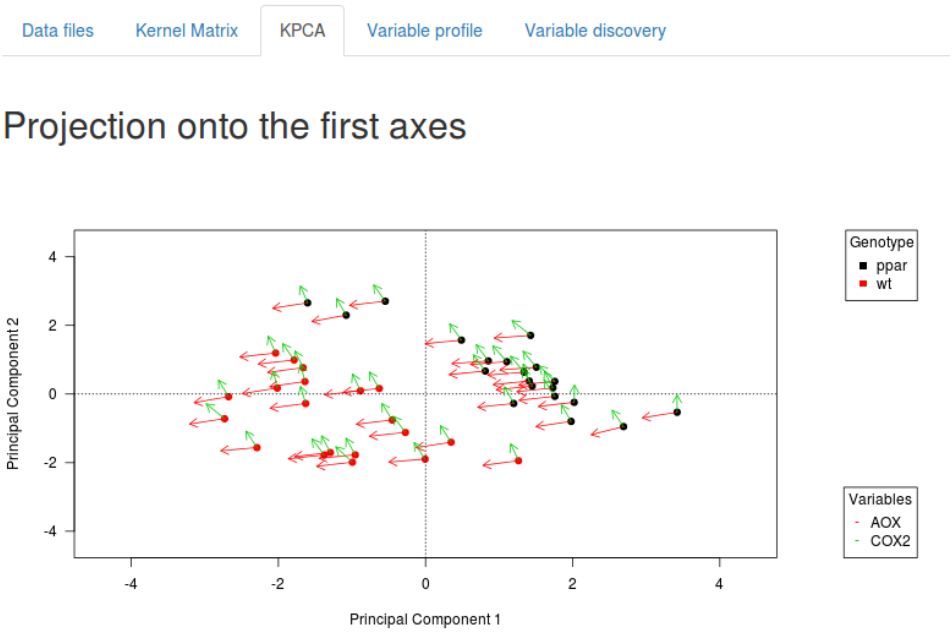

Figura 11: *Kernel* PCA amb representació de variables originals.

Variable profile

En cas de representar variables en el *kernel* PCA, aquesta pestanya ens permetrà veure el perfil de cada variable que es seleccioni (Figura 12).

Variable discovery

Donat un vector d'interès en el *kernel* PCA, obtindrem el llistat de variables amb el valor mitjà de la correlació i la desviació estàndard. Addicionalment es proporciona una representació gràfica dels resultats (Figura 13).

Finalment, s'ha creat un tutorial en HTML per tal de guiar els usuaris en l'anàlisi de dades mitjançant aquesta aplicació (Figura 14). Aquest petit manual exemplifica pas a pas, en un conjunt de dades d'exemple, un anàlisi de components principals amb *kernels*. Mostra com definir el *kernel* i els paràmetres del *kernel* a utilitzar, com visualitzar diferents variables o combinacions de variables i com cercar variables d'interès.

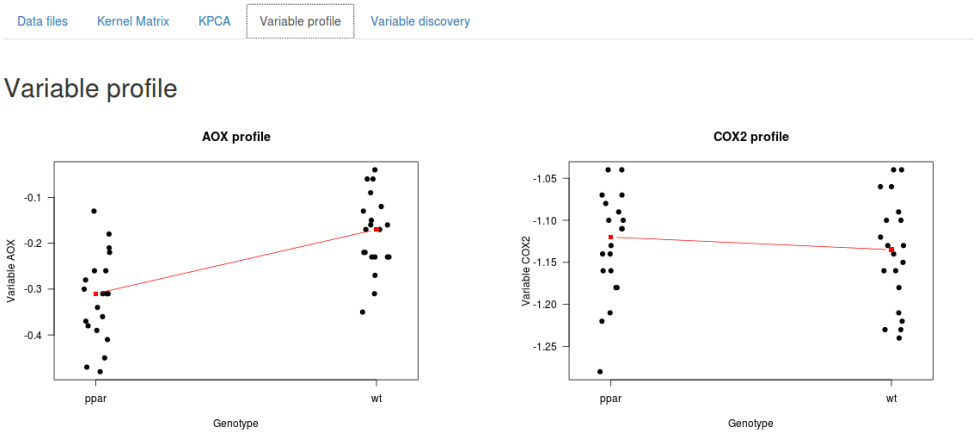

Figura 12: Representació del perfil de les variables seleccionades.

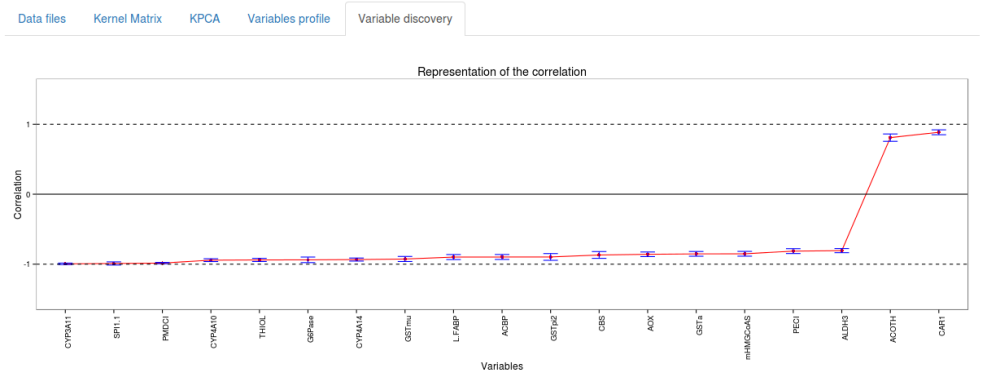

Figura 13: Representació gràfica de les variables més correlacionades amb el vector d'interès definit en el KPCA.

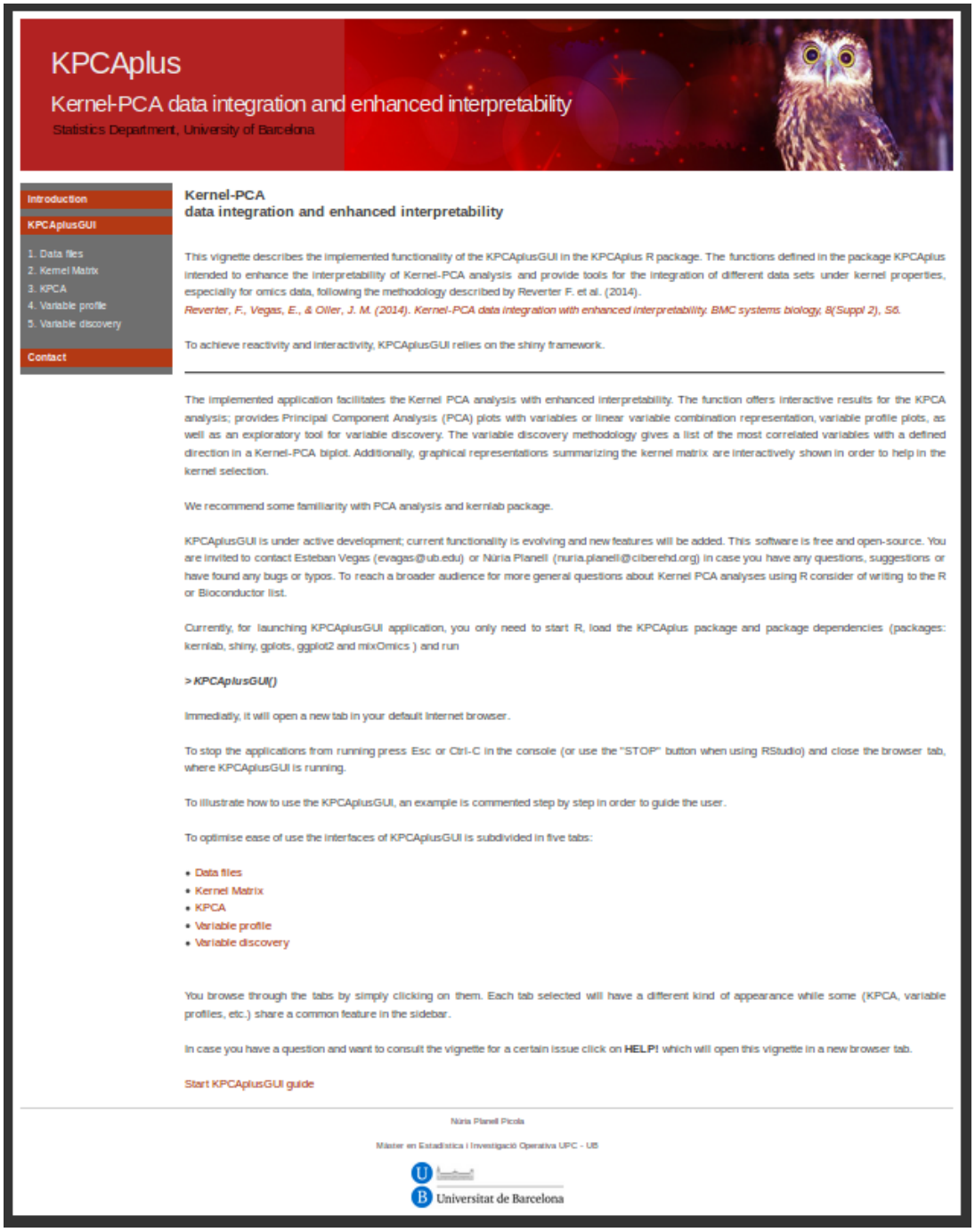

Figura 14: Documentació de l'aplicació creada (`KPCAplusGUI`) en HTML.

4. Llibreria `KPCAplus`

Seguint la filosofia de R, s'ha creat la llibreria `KPCAplus` que engloba la funció `KPCAplus` i la seva aplicació web `KPCAplusGUI`, per poder compartir-la fàcilment amb altres usuaris interessats [27]. La creació de l'aplicació web de la funció `KPCAplus` ens proporciona una major divulgació del programari fent-lo accessible tant a usuaris poc familiaritzats amb R com a usuaris experts.

Així, els usuaris podran investigar la integració de dades amb kernels, la visualització de variables o combinacions de variables i el descobriment de variables d'interès. Dins del programari lliure, un punt molt important i característic d'aquest, és el *feedback* dels usuaris que ajudarà a millorar i validar la utilitat d'aquests mètodes.

La llibreria es proporciona en format *"package source"* (KPCAplus_1.0.tar.gz) i per instal·lar-la tan sols cal iniciar R i instal·lar la llibreria.

```
> install.packages("KPCAplus_1.0.tar.gz")
```

Un cop carregada, per executar la funció o l'aplicació web podrem fer-ho com segueix:

```
> library("KPCAplus")
> # Documentació de la funció KPCAplus
> help(KPCAplus)
> # Executem la funció
> KPCAplus(x=dataset, kernel_function="rbfdot",
> + kparameters=list(sigma=0.05), factors=phenotypic_data,
> + main="Kernel PCA")
> # Documentació de la funció que executa l'applicació web
> help(KPCAplusGUI)
> # Execusió applicació web
> KPCAplusGUI()
```

Capítol 3.
Aplicació en dades reals

1. Malaltia inflamatòria intestinal

La malaltia inflamatòria intestinal (MII) representa un grup d'afectacions intestinas idiopàtiques de caràcter crònic, destacant-ne la malaltia de Crohn i la colitis ulcerosa. L'etiologia d'aquestes és desconeguda, no obstant, s'han descrit factors genètics i ambientals, com la modificació dels bacteris luminals i l'augment de la permeabilitat intestinal, que poden jugar un paper rellevant en la regulació de la immunitat intestinal i, conseqüentment, en l'aparició de lesions a la mucosa.
La incidència de la MII és de 3-15 nous casos en la colitis ulcerosa i de 1-10 nous casos en la malaltia de Crohn per 100.000 habitants/any [28, 29].

1.1. Colitis Ulcerosa

La colitis ulcerosa (CU) és una malaltia inflamatòria intestinal que pot afectar tot el còlon, des del recte fins al cec de forma contínua (Figura 15 A). Es defineix l'activitat de la colitis ulcerosa en funció de la inflamació de la mucosa intestinal, que es dóna en forma d'eritema, disminució o pèrdua del patró vascular, friabilitat, erosions i presència d'úlceres, depenent de la gravetat. Aquesta desestructuració de la mucosa intestinal s'associa a diversos símptomes clínics com són la diarrea, el sagnat al defecar, el dolor abdominal, la pèrdua d'apetit, la pèrdua de pes i el cansament, entre altres [28, 30]. Al ser una malaltia crònica, aquesta es caracteritza per presentar, des del seu diagnòstic, períodes d'activitat seguits de períodes de remissió de la malaltia (Figura 15 B).

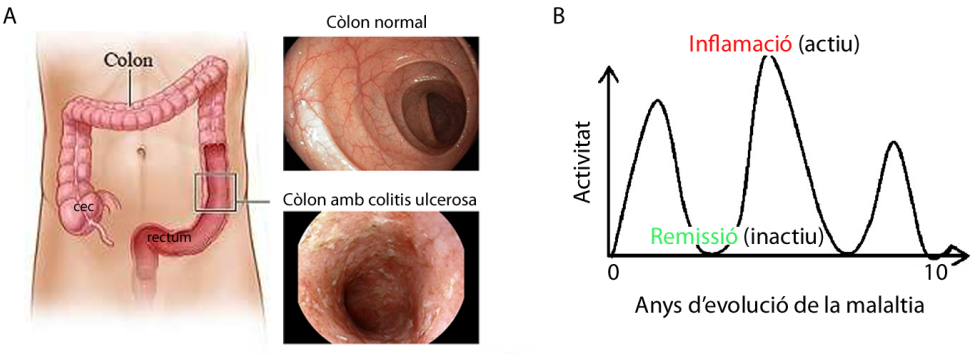

Figura 15: Colitis ulcerosa. A) La colitis ulcerosa presenta una afectació confinada al còlon, podent afectar de forma contínua des del recte fins al cec. Les imatges ampliades exemplifiquen la imatge endoscòpica d'una mucosa sana i la d'una mucosa afectada per colitis ulcerosa. B) Esquema de l'evolució dels malalts amb colitis ulcerosa al llarg del temps, intercalant períodes d'activitat i remissió que es succeeixen des del moment del diagnòstic de la malaltia.

1.2. Mucosa intestinal

La mucosa intestinal està formada per diversos tipus cel·lulars que ajuden a mantenir l'homeòstasi del còlon, entre els que trobem les cèl·lules epitelials, les cèl·lules del sistema immunitari, els fibroblasts i els bacteris luminals. La mucosa es confina a la part interna del còlon, pel que està en contacte directe amb la llum intestinal on trobarem els bacteris comensals i els nutrients provinents de la dieta.

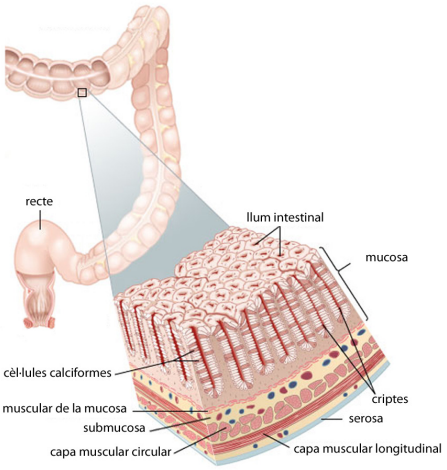

Figura 16: Esquema histològic del còlon.

Tal i com s'il·lustra a la Figura 16, la mucosa s'estructura en criptes subjectes a la làmina pròpia, i per sota trobem vàries capes; la muscular de la mucosa, la submucosa, les capes musculars circulars i longitudinals i, finalment, la serosa, que donaran estructura i mobilitat al còlon.
La monocapa de cèl·lules epitelials que formen les criptes de la mucosa intestinal representa la primera línia de defensa contra els agents externs (patògens). Està formada per diferents tipus cel·lulars, destacant-ne la funció de les cèl·lules que s'encarreguen de la secreció de moc i de proteïnes antimicrobianes que promouen l'eliminació bacteriana de la superfície epitelial. Alteracions en la barrera intestinal, com un augment de la permeabilitat o una disminució del moc, podran conduir a la migració de bacteris comensals i productes microbians de la llum intestinal a l'interior de la capa mucosa, iniciant una resposta immunitària. La resposta immunitària activarà una complexa xarxa de mecanismes per a defensar l'organisme de l'agressió dels agents externs, actuant mitjançant la secreció de nombroses citoquines que afavoriran el reclutament de les cèl·lules immunes (neutròfils, macròfags, limfòcits, etc.) per a l'eliminació dels elements patogènics que hagin irromput a l'interior de la mucosa intestinal. El bon funcionament de tots aquests elements és primordial per a mantenir l'homeòstasi intestinal (Figura 17) pel que, alteracions en la composició de la flora microbiana, un augment de la permeabilitat intestinal, canvis en el balanç del sistema enteroendocrí i una desregulació del sistema immunitari poden representar factors clau en el desenvolupament de la malaltia inflamatòria intestinal. No obstant, el procés exacte que condueix a l'aparició de la malaltia encara és desconegut [35–37].

2. Estudi transcripcional

Varis estudis transcripcionals s'han realitzat per tal de dilucidar els mecanismes subjectes a la desregulació biològica de la malaltia, identificar biomarcadors i descobrir noves dianes terapèutiques [31–33]. Entre aquests, trobem un estudi publicat fa un parell d'anys pel grup d'investigació en malaltia inflamatòria intestinal de l'Hospital Clínic i Provincial de Barcelona del que formo part, que posa en relleu, mitjançant l'anàlisi transcripcional de mostres de pacients amb colitis ulcerosa, les alteracions transcripcionals associades a la mucosa intestinal en els diferents estadis de la malaltia (activitat versus remissió) [34].

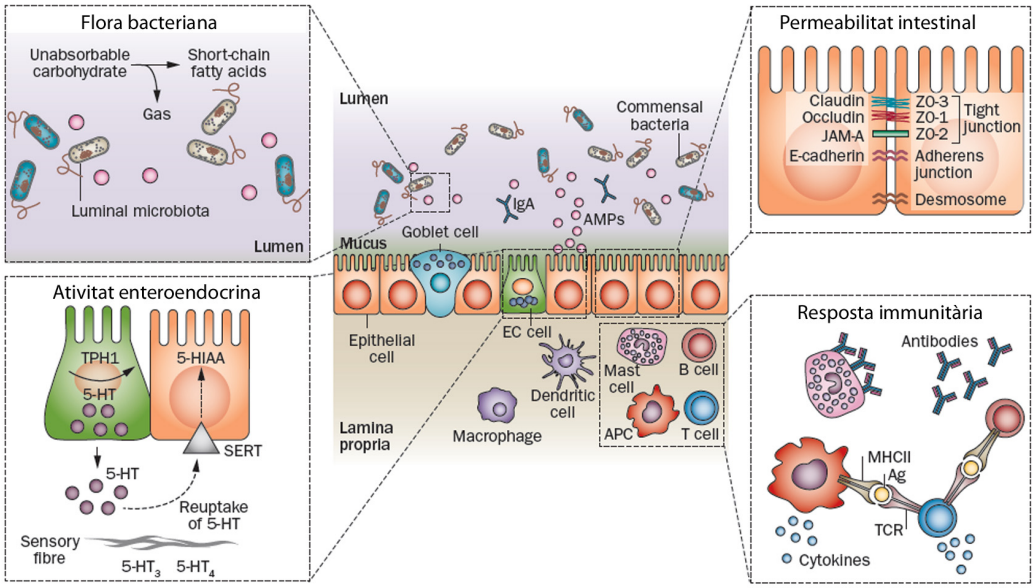

Figura 17: La barrera intestinal. Les cèl·lules epitelials de l'intestí formen una barrera bioquímica i física entre la llum intestinal i el sistema immunitari de la mucosa. El manteniment de l'homeòstasi intestinal davant de la constant exposició als bacteris comensals s'aconsegueix gràcies a varis factors que treballen de forma coordinada, englobant la permeabilitat intestinal, l'activitat enteroendocrina, la resposta del sistema immunitari de la mucosa i, al mateix temps, la composició microbiana que es troba a la llum intestinal [35].

Amb l'objectiu de determinar els canvis transcripcionals associats a la mucosa intestinal en remissió de pacients amb colitis ulcerosa, es van analitzar biòpsies d'individus sans, de pacients amb colitis ulcerosa en fase activa i de pacients amb colitis ulcerosa en remissió. En total es van incloure 89 individus, que van dividir-se en tres grups independents; 36 individus per a l'anàlisi transcripcional amb *microarrays*, 30 individus per a la validació transcripcional per *real-time* RT-PCR i 23 individus per a la validació a nivell proteic mitjançant tècniques d'immunohistoquímica i immunofluorescència.

2.1. Anàlisi de *microarrays*

Centrant-nos en l'anàlisi de *microarrays*, es van processar 43 mostres corresponents a 36 individus. Tretze mostres es van obtenir de controls sans, 8 mostres de mucosa "curada" de pacients amb colitis ulcerosa en remissió (UCI), 15 mostres de mucosa inflamada de pacients amb colitis ulcerosa activa (UCA+) i, d'aquests 15, en 7 casos també es va agafar mostra de mucosa sana (UCA-) (Figura 18).

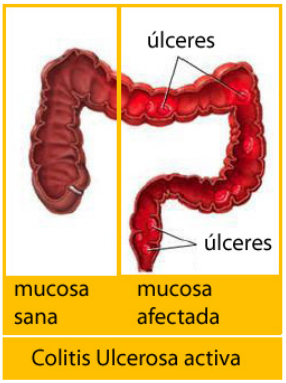

Figura 18: Localització de les mostres. A l'esquerra es mostra el que seria una biòpsia d'una zona sana d'un pacient amb colitis ulcerosa actiu (UCA-). A la dreta es mostra una zona afectada (inflamada) del mateix pacient actiu (UCA+).

Es van utilitzar *microarrays* comercials d'Affymetrix (Affymetrix Gene Chip Human Genome U133 Plus 2.0 Arrays, 54.675 sondes), i les dades crues resultants (fitxers .CEL) estan disponibles al repositori per a dades genòmiques de NCBI, *Gene Expression Omnibus* (GEO), amb el codi d'accés GSE38713.

L'anàlisi transcripcional d'aquestes dades va revelar un perfil transcripcional característic de la mucosa intestinal en remissió de pacients amb colitis ulcerosa.
La inspecció de les dades mitjançant l'anàlisi de components principals mostra una clara segregació de les mostres en tres grups, les mostres de zones inflamades de pacients, les mostres de zones "curades" de pacients en remissió de la malatia i les mostres de controls sans o de les zones sanes de pacients amb colitis ulcerosa (Figura 19). Posteriorment, l'anàlisi d'expressió gènica diferencial utilitzant els models lineals per a l'anàlisi de *microarrays* (LIMMA) va permetre identificar aquells gens característics de cada estadi de la malaltia. Definint un p-valor corregit (FDR) < 0.05 i una taxa de canvi en valor absolut > 1.5, s'identifiquen un total de 6.365 gens diferencialment expressats entre la mucosa de controls sans i els pacients actius. D'aquests, es detecten 3.700 gens que es mantenen alterats en la mucosa en remissió. L'anàlisis comparatiu entre la mucosa de controls sans respecte la mucosa sana dels pacients amb colitis ulcerosa no mostra alteracions significatives a nivell transcripcional.

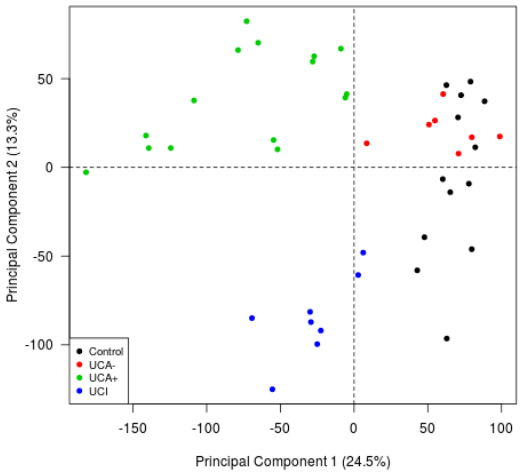

Figura 19: Anàlisi de components principals (PCA) de dades transcripcionals derivades de mostres de biòpsies intestinals humanes. Les mostres de mucosa intestinal es van obtenir de controls sans (Controls, en negre, n=13), de biòpsies sanes de pacients amb colitis ulcerosa activa (UCA-, en verd, n=7), de biòpsies inflamades de pacients amb colitis ulcerosa activa (UCA+, en vermell, n=15), i de biòpsies "curades" de pacients amb colitis ulcerosa en remissió (UCI, en blau, n=8).

Entre els gens identificats com a sobre-expressats en la mucosa inflamada de pacients amb colitis ulcerosa activa identifiquem IL-8, REG1A, CXCL3 i IL-1B, entre molts d'altres, que codifiquen molècules pro-inflamatòries involucrades en la resposta immunològica. Els gens de la IL-8, la IL-1B i el CXCL3 intervenen en el reclutament de neutròfils, mentre que REG1A, tot i que la funció exacta no es coneix, s'ha descrit en la inflamació epitelial i en els darrers anys s'ha relacionat amb la metaplàsia de les cèl·lules de Paneth característica de la mucosa intestinal dels pacients amb MII [37, 38].

Aquestes senyals pro-inflamatòries es perden en la mucosa intestinal en remissió. No obstant, existeix un subconjunt de gens que es mantenen alterats en la mucosa dels pacients amb colitis ulcerosa en remissió. Alguns d'aquests són REG4, S100P, AQP3, AQP8 i ABCG2, mostrant nivells transcripcionals alterats tant en la mucosa inflamada com a la mucosa en remissió en comparació amb la mucosa sana. Aquests gens són expressats principalment per les cèl·lules epitelials, indicant-nos que aquesta mucosa no torna a la normalitat després d'un brot de la malaltia.

3. *Kernel* PCA per a estudis transcripcionals

Prenent com a referència els resultats de l'estudi de la colitis ulcerosa en remissió, a continuació s'exploren les dades d'aquest estudi utilitzant el mètode *kernel* PCA amb visualització de variables, emprant la llibreria en R que s'ha desenvolupat.

3.1. Elecció de la funció *kernel*

El primer pas en el *kernel* PCA és la determinació de la funció *kernel* i els paràmetres a utilitzar. Treballem amb el conjunt de dades transcripcionals normalitzades i filtrades, que conté la informació de 22.206 sondes per 43 observacions. La Figura 20 mostra el *kernel* PCA resultant d'aplicar un *kernel* gaussià amb un valor de $\sigma = 0.0001$.

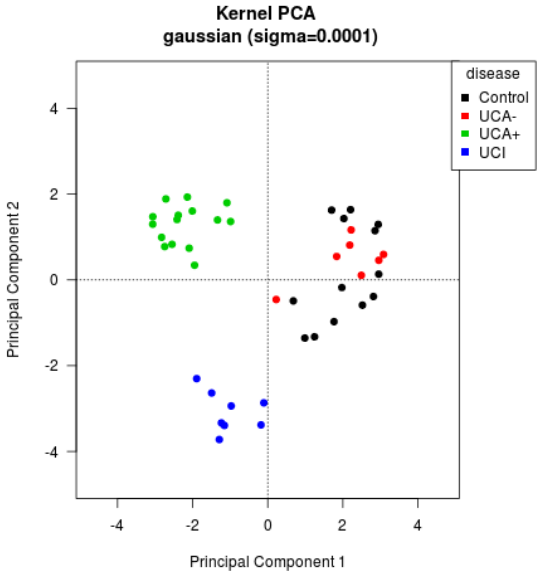

Figura 20: *Kernel* PCA del conjunt de dades transcripcionals derivades de mostres de biòpsies intestinals humanes aplicant un *kernel* gaussià amb $\sigma = 0.0001$. Cada punt representa una observació (biòpsia). Les mostres de mucosa intestinal es van obtenir de controls sans (Controls, en negre, n=13), de biòpsies sanes de pacients amb colitis ulcerosa activa (UCA-, en verd, n=7), de biòpsies inflamades de pacients amb colitis ulcerosa activa (UCA+, en vermell, n=15), i de biòpsies "curades" de pacients amb colitis ulcerosa en remissió (UCI, en blau, n=8).

A partir del *kernel* seleccionat, podem observar com el PCA resultant d'aplicar un *kernel* gaussià amb $\sigma = 0.0001$ distribueix les mostres en tres clústers, essent la segregació encara més evident que en el resultat del PCA lineal (Figura 19).

3.2. Representació de variables originals

Definit el *kernel*, el següent pas serà l'exploració de les variables originals en aquest nou espai de dimensió reduïda. Fixant-nos en aquells gens que s'han descrit i validat en l'estudi, s'avaluarà la representació gràfica d'aquests en el *kernel* PCA esperant observar un patró similar. Estudiarem diferents situacions, visualització de gens pro-inflamatoris, de gens alterats en la mucosa de pacients amb colitis ulcerosa ("curada" o inflamada), de gens característics de la mucosa "curada" i de gens constitutius (*housekeeping genes*). Addicionalment, es visualitzarà la combinació lineal d'alguns gens que formen dímers.

- Gens pro-inflamatoris

  Els gens pro-inflamatoris són aquells relacionats amb l'activació de la inflamació. Davant de qualsevol procés inflamatori que es doni a l'organisme trobarem elevats els nivells d'aquests gens, que estan associats a diferents tipus cel·lulars del sistema immunitari encarregats de regular aquest procés. En les dades estudiades, esperarem observar un augment d'aquests en la mucosa inflamada dels pacients amb colitis ulcerosa activa. A continuació, es mostra la representació d'alguns gens pro-inflamatoris en el *kernel* PCA generat anteriorment, juntament amb el perfil transcripcional corresponent.

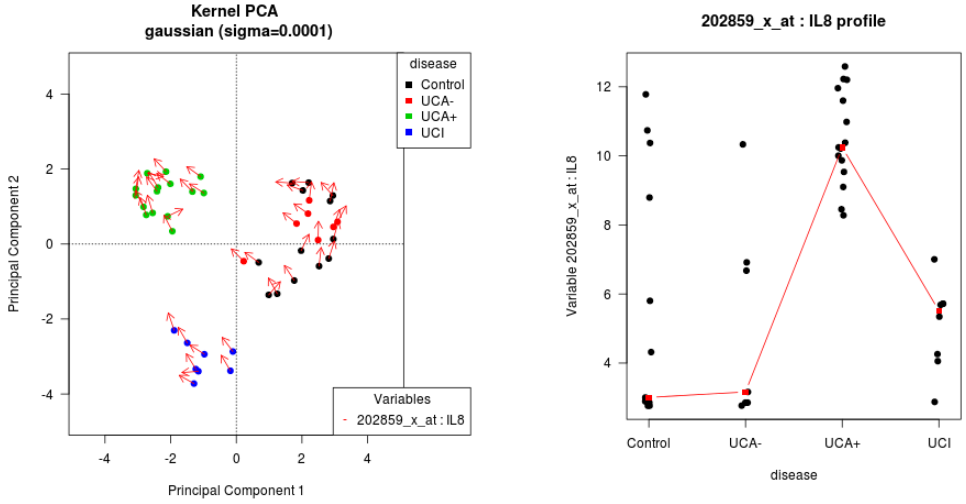

Figura 21: A l'esquerra es mostra el *kernel* PCA de l'anàlisi transcripcional derivat de mostres de biòpsies intestinals humanes aplicant un kernel gaussià amb $\sigma = 0.0001$ amb representació de variables. Les fletxes en vermell representen el gen IL8. A la dreta es mostra el perfil transcripcional del gen IL8 per cada grup de mostres.

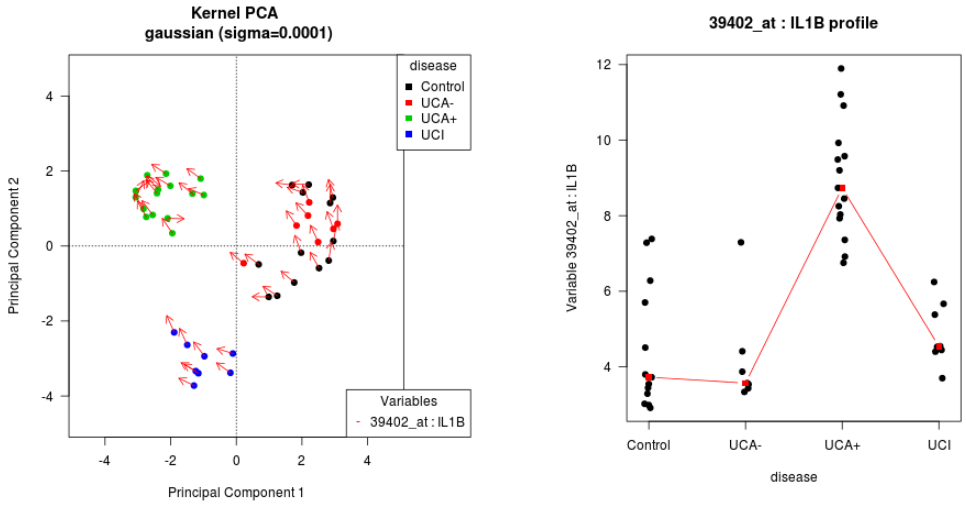

Figura 22: A l'esquerra es mostra el *kernel* PCA de l'anàlisi transcripcional derivat de mostres de biòpsies intestinals humanes aplicant un kernel gaussià amb $\sigma = 0.0001$ amb representació de variables. Les fletxes en vermell representen el gen IL1B. A la dreta es mostra el perfil transcripcional del gen IL1B per cada grup de mostres.

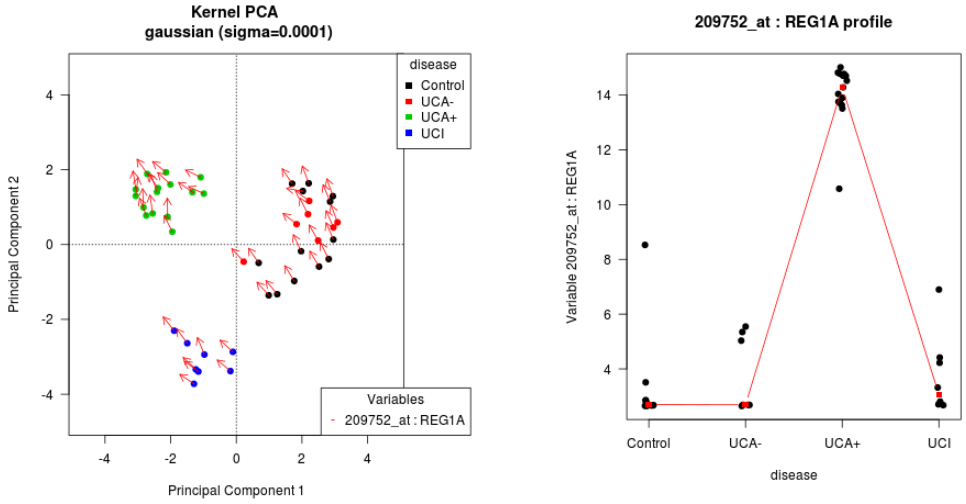

Figura 23: A l'esquerra es mostra el *kernel* PCA de l'anàlisi transcripcional derivat de mostres de biòpsies intestinals humanes aplicant un kernel gaussià amb $\sigma = 0.0001$ amb representació de variables. Les fletxes en vermell representen el gen REG1A. A la dreta es mostra el perfil transcripcional del gen REG1A per cada grup de mostres.

Fixant-nos en els perfils transcipcionals podem veure com IL8, IL1B i REG1A estan alterats en la mucosa inflamada dels pacients amb colitis ulcerosa activa, com s'havia descrit anteriorment. Aquest resultat també es pot deduir de la visualització d'aquestes variables en el *kernel* PCA. En tots tres casos apareix una direcció clara de les variables cap al grup de mostres representades en color verd, corresponent a les biòpsies de pacients actius (Figura 21, 22, 23).

- Gens alterats en la mucosa de pacients amb colitis ulcerosa
  Aquest grup de gens engloba aquells que presenten patrons transcripcionals alterats en la mucosa dels pacients amb colitis ulcerosa (inflamada o "curada"), tant en pacients en fase activa com en pacients en remissió de la malaltia. Gran part d'aquests gens són expressats per cèl·lules epitelials, com AQP8 i ABCG2, que codifiquen proteïnes de membrana relacionades amb el transport cel·lular, i REG4 i S100P, que s'han descrit en alguns tumors epitelials com el de càncer colorectal.

  El perfil transcripcional de AQP8 i ABCG2 mostra una disminució clara d'aquests en la mucosa activa o inactiva dels pacients amb colitis ulcerosa respecte els controls sans. La representació de variables en el *kernel* PCA ens revela com les fletxes d'ambdós gens apunten cap al grup d'individus control i mostres sanes de pacients (punts negres i vermells, respectivament), coincidint amb el comportament descrit (Figura 24, 25).
  Per altra banda, el perfil transcripcional dels gens amb un patró oposat, com S100P i REG4, presenten un augment de l'expressió transcripcional en la mucosa de pacients amb colitis ulcerosa activa o en remissió. La representació de variables en el *kernel* PCA ens permet veure la direcció a la que apunten aquestes variables, identificant una tendència cap a les mostres afectades d'individus amb colitis ulcerosa (punts verds i blaus)(Figura 26, 27).

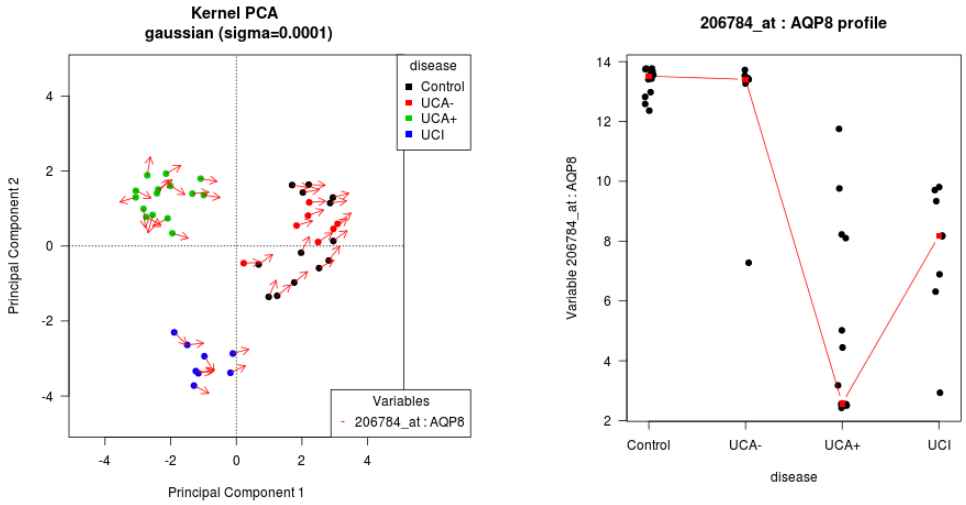

Figura 24: A l'esquerra es mostra el *kernel* PCA de l'anàlisi transcripcional derivat de mostres de biòpsies intestinals humanes aplicant un kernel gaussià amb $\sigma = 0.0001$ amb representació de variables. Les fletxes en vermell representen el gen AQP8. A la dreta es mostra el perfil transcripcional del gen AQP8 per cada grup de mostres.

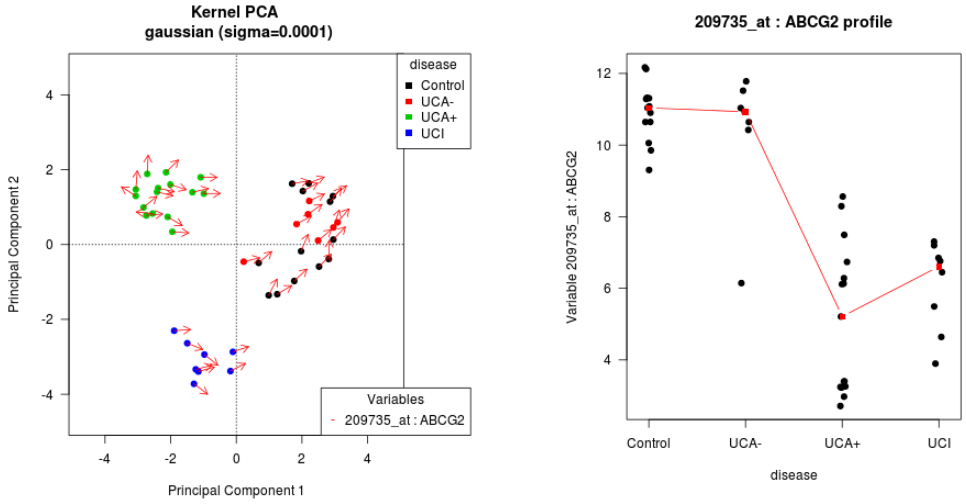

Figura 25: A l'esquerra es mostra el *kernel* PCA de l'anàlisi transcripcional derivat de mostres de biòpsies intestinals humanes aplicant un kernel gaussià amb $\sigma = 0.0001$ amb representació de variables. Les fletxes en vermell representen el gen ABCG2. A la dreta es mostra el perfil transcripcional del gen ABCG2 per cada grup de mostres.

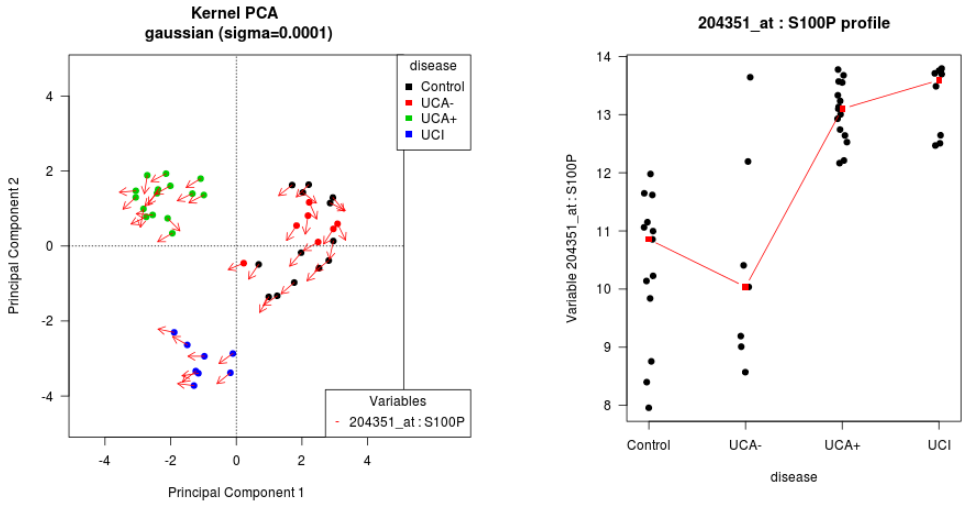

Figura 26: A l'esquerra es mostra el *kernel* PCA de l'anàlisi transcripcional derivat de mostres de biòpsies intestinals humanes aplicant un kernel gaussià amb $\sigma = 0.0001$ amb representació de variables. Les fletxes en vermell representen el gen S100P. A la dreta es mostra el perfil transcripcional del gen S100P per cada grup de mostres.

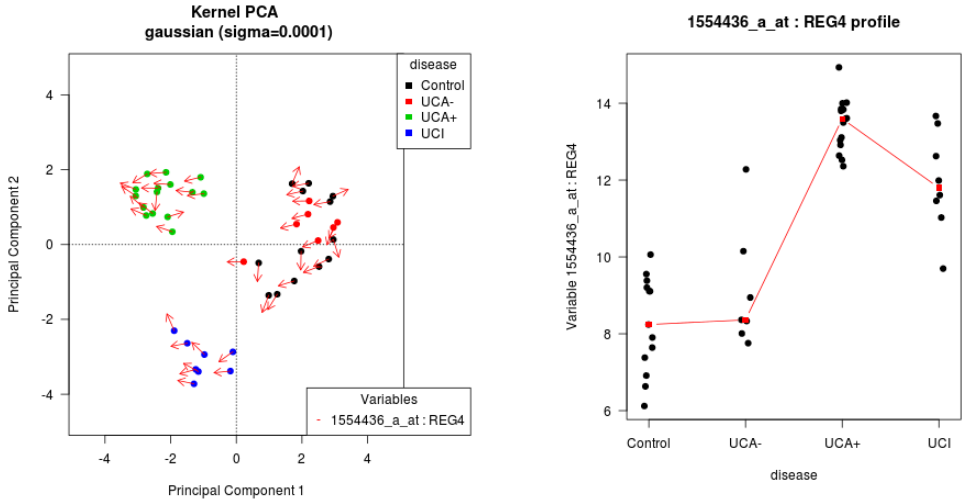

Figura 27: A l'esquerra es mostra el *kernel* PCA de l'anàlisi transcripcional derivat de mostres de biòpsies intestinals humanes aplicant un kernel gaussià amb $\sigma = 0.0001$ amb representació de variables. Les fletxes en vermell representen el gen REG4. A la dreta es mostra el perfil transcripcional del gen REG4 per cada grup de mostres.

- Gens característics de la mucosa "curada"
  Un altre grup de gens molt interessant d'investigar és el que inclou aquells gens diferencialment expressats exclusivament en la mucosa dels pacients amb colitis ulcerosa en remissió. Tot i que l'article publicat no entra en aquesta aproximació, l'anàlisi d'expressió diferencial de les dades ens permet identificar quins són aquests gens, trobant-ne un total de 1.647.
  GPR128 és un dels gens que veiem alterats, i que s'ha relacionat amb la pèrdua de pes i l'increment de la freqüència de contracció intestinal en ratolins mutats en els que s'ha eliminat aquest gen [39]. En les mostres de biòpsies humanes observem com el perfil transcripcional de GPR128 disminueix en els pacients amb colitis ulcerosa en remissió, coincidint amb el que indica el resultat de la representació de la variable en el *kernel* PCA, on la variable apunta a la direcció contrària d'on es situen aquestes mostres (Figura 28).

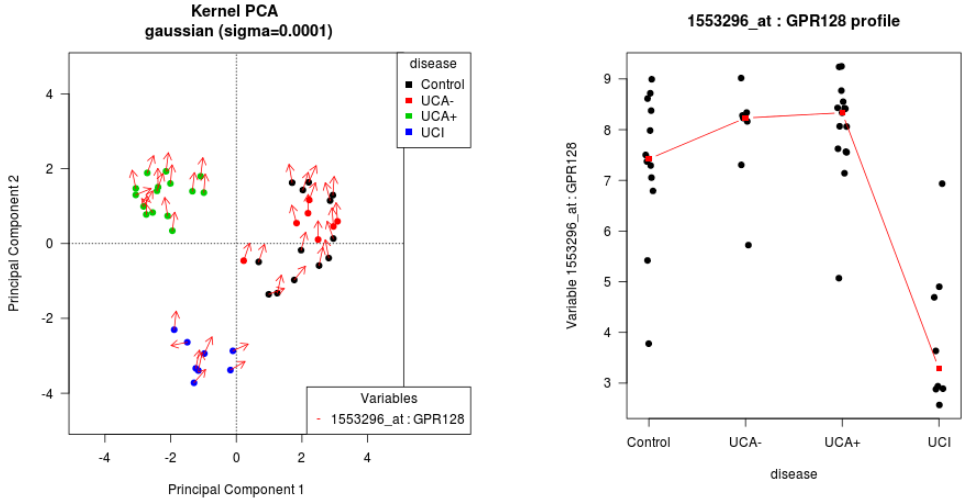

Figura 28: A l'esquerra es mostra el *Kernel* PCA de l'anàlisi transcripcional derivat de mostres de biòpsies intestinals humanes aplicant un kernel gaussià amb $\sigma = 0.0001$ amb representació de variables. Les fletxes en vermell representen el gen GPR128. A la dreta es mostra el perfil transcripcional del gen GPR128 per cada grup de mostres.

- Gens constitutius (*housekeeping genes*)

  Els gens constitutius o *housekeepings* són aquells gens necessaris pel manteniment de les funcions cel·lulars bàsiques, expressant-se de forma constant en totes les cèl·lules i condicions d'un organisme [40]. Si inspeccionem el perfil transcripcional d'un d'aquests gens, com UBA3, observem que no existeixen diferències transcripcionals entre els diferents grups de mostres i, fixant-nos en la representació d'aquest en el *kernel* PCA, no identifiquem una direcció clara cap on apuntin les fletxes (Figura 29).

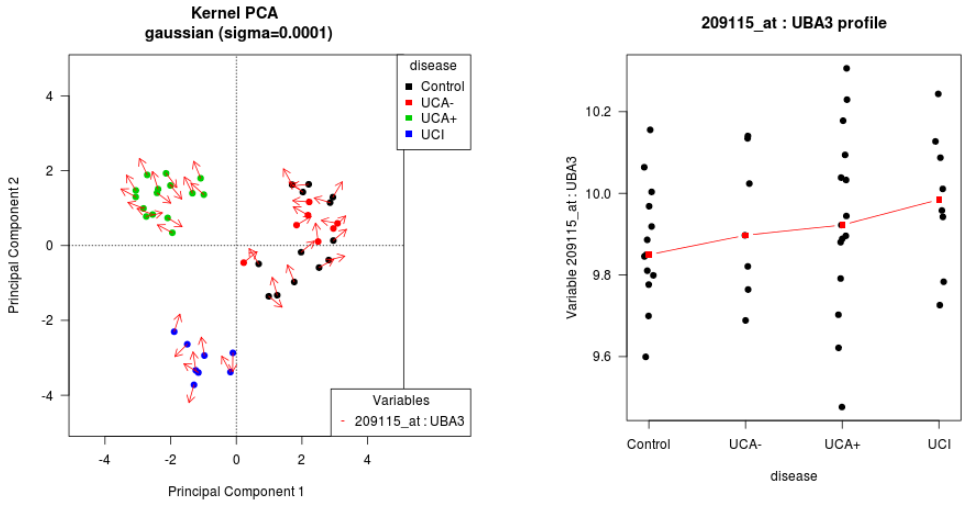

Figura 29: A l'esquerra es mostra el *kernel* PCA de l'anàlisi transcripcional derivat de mostres de biòpsies intestinals humanes aplicant un kernel gaussià amb $\sigma = 0.0001$ amb representació de variables. Les fletxes en vermell representen el gen UBA3. A la dreta es mostra el perfil transcripcional del gen UBA3 per cada grup de mostres.

- Combinació linial de gens

  En la natura trobem proteïnes formades per més d'una molècula, com la calprotectina, una proteïna secretada per les cèl·lules del sistema immunològic (principalment neutròfils) i que s'ha descrit com a marcador inflamatori [41]. La codifiquen els gens S100A8 i S100A9, pel que en les nostres dades d'estudi, esperaríem veure una expressió augmentada d'ambdós en les mostres de mucosa inflamada de pacients amb colitis ulcerosa. La representació de la combinació lineal d'aquests dos gens en el *kernel* PCA apunta clarament cap al grup de mostres inflamades (Figura 30, punts verds), i el perfil transcripcional de cada gen per separat també concorda amb el que s'ha descrit (Figura 31).

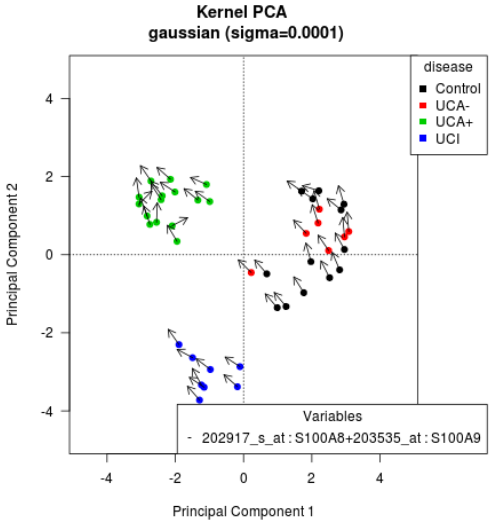

Figura 30: *Kernel* PCA de l'anàlisi transcripcional derivat de mostres de biòpsies intestinals humanes aplicant un kernel gaussià amb $\sigma = 0.0001$ amb representació de variables. La fletxa negra mostra la combinació lineal dels gens S100A8 i S100A9.

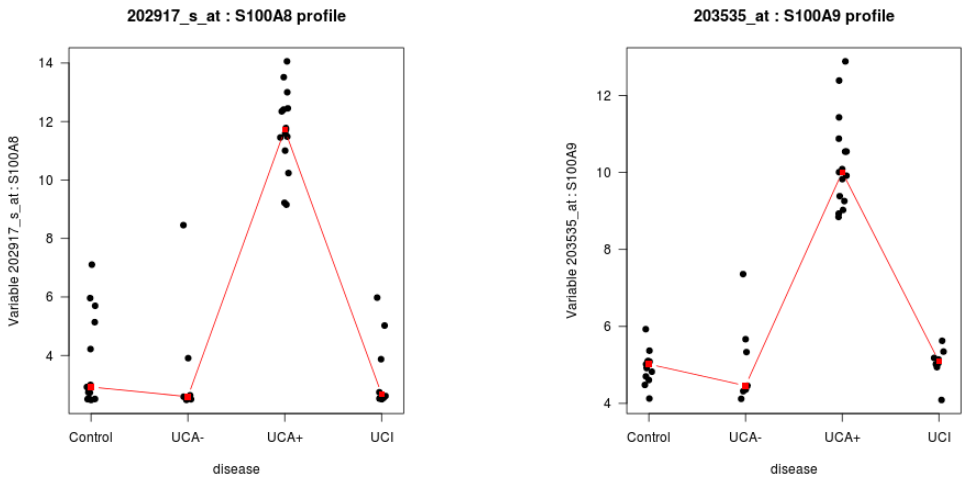

Figura 31: Perfil transcripcional dels gens S100A8 (esquerra) i S100A9 (dreta) per cada grup de mostres.

3.3. Cerca de variables d'interès

Avaluada la concordança entre els resultats publicats i la representació de variables en el *kernel* PCA, a continuació, mitjançant l'estratègia per a la cerca de variables d'interès implementada en la llibreria, s'exploren:

- Les variables correlacionades amb el vector que apunta en direcció a les mostres de mucosa inflamada de pacients actius (Figura 32, gràfic de l'esquerra).
- Les variables correlacionades amb el vector que apunta en direcció a les mostres de mucosa de pacients en remissió (Figura 32, gràfic de la dreta).

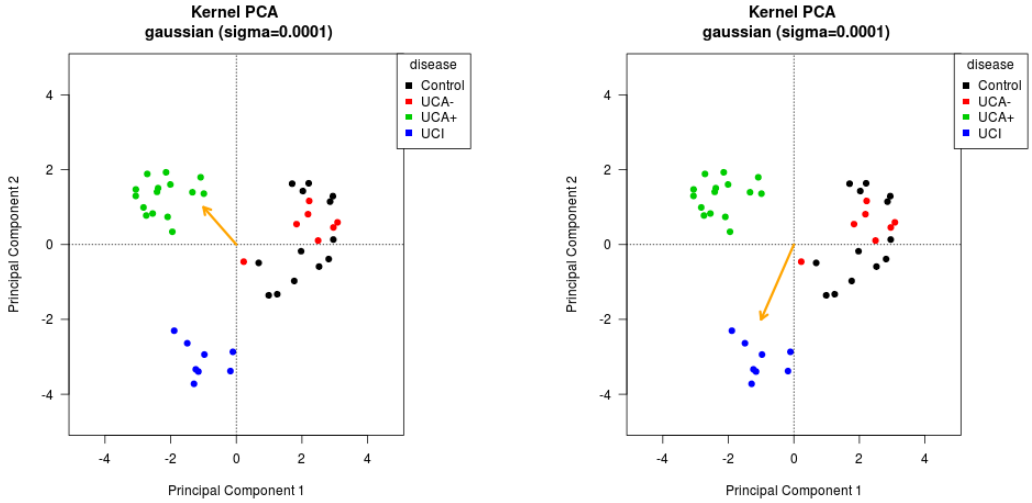

Figura 32: Cerca de variables en el *kernel* PCA. Al gràfic de l'esquerra es defineix un vector que apunta al clúster de mostres de mucosa inflamada de pacients (punt:$-1, 1$). Al gràfic de la dreta es defineix un vector que apunta al clúster de mostres de mucosa en remissió (punt: $-1, -2$).

Definit el vector que apunta a $(-1, 1)$ i un valor mitjà de correlació $|r| \geq 0.8$ amb una desviació estàndard inferior a la corresponent al primer quartil, identifiquem 204 sondes estretament correlacionades amb aquest vector (Taula 1). Per tal de fer-nos una idea de quin és el comportament d'aquestes segons l'anàlisi d'expressió diferencial que s'acostuma a fer quan es treballa amb dades transcripcionals de *microarrays*, s'han representat les 204 sondes segons la taxa de canvi (log2-fold change) i el p-valor corregit (FDR) obtinguts de comparar les mostres de mucosa inflamada de pacients amb colitis ulcerosa respecte les mostres dels controls sans (Figura 33, gràfic de l'esquerra).

Gen	Mitjana	Desviació estàndard
205749_at : CYP1A1	-0.83	0.27
208904_s_at : RPS28	-0.83	0.34
211018_at : LSS	-0.82	0.32
225275_at : EDIL3	-0.82	0.31
⋮	⋮	⋮
205941_s_at : COL10A1	0.94	0.30
213832_at : KCND3	0.94	0.09
209752_at : REG1A	0.94	0.07
205886_at : REG1B	0.95	0.12

Table 1: Resum de les variables més correlacionades amb el vector que apunta a $(-1, 1)$.

Podem observar com gran part de les variables identificades també es detecten utilitzant l'anàlisi habitual, com REG1A, vist anteriorment (Figura 23). No obstant, 50 d'aquests gens es descarten en els anàlisis d'expressió diferencial per presentar taxes de canvi inferiors a 1.5 i un p-valor corregit (FDR)$>$ 0.05 i només se n'identifiquen 7 amb una expressió diferencial disminuida.

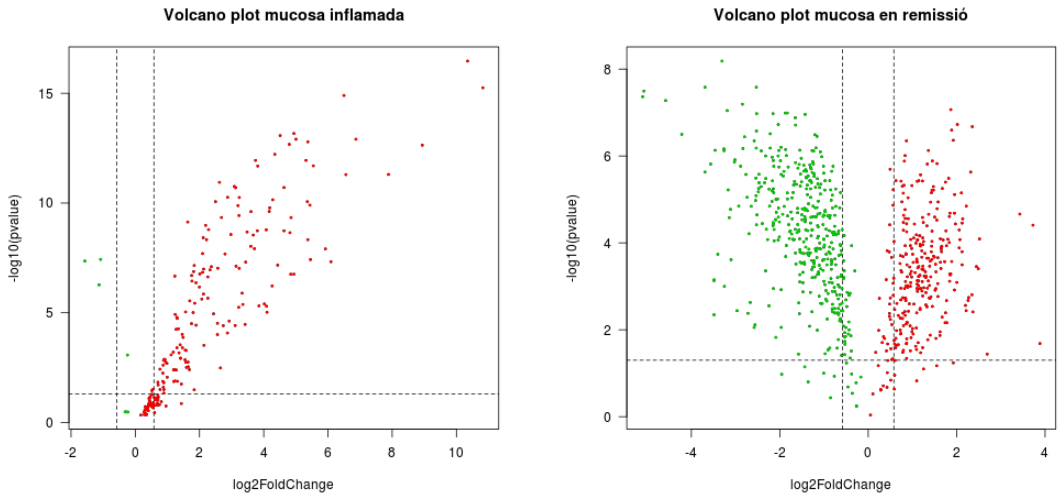

Figura 33: *Volcano plot* amb l'eix de les x mostrant el logaritme en base 2 de la taxa de canvi i l'eix de les y el negatiu del logaritme en base 10 del p-valor ajustat. Al gràfic de l'esquerra es mostren les variables correlacionades amb el punt $(-1, 1)$. Al gràfic de la dreta es mostren les variables correlacionades amb el punt $(-1, -2)$.

Si determinem aquells gens característics de la mucosa inflamada de pacients amb colitis ulcerosa activa mitjançant l'anàlisi d'expressió diferencial amb els criteris definits anteriorment, identifiquem un total de 2.963 sondes corresponents a 2.182 gens (Figura 34 A), mentre que l'estudi de correlació de les variables amb la direcció que hem definit només identifica 204 sondes corresponents a 172 gens. Així, el nombre de variables que identifiquem com a rellevants és molt menor utilitzant el mètode de correlació; no obstant, el 67% d'aquestes són comuns amb l'anàlisi d'expressió diferencial.

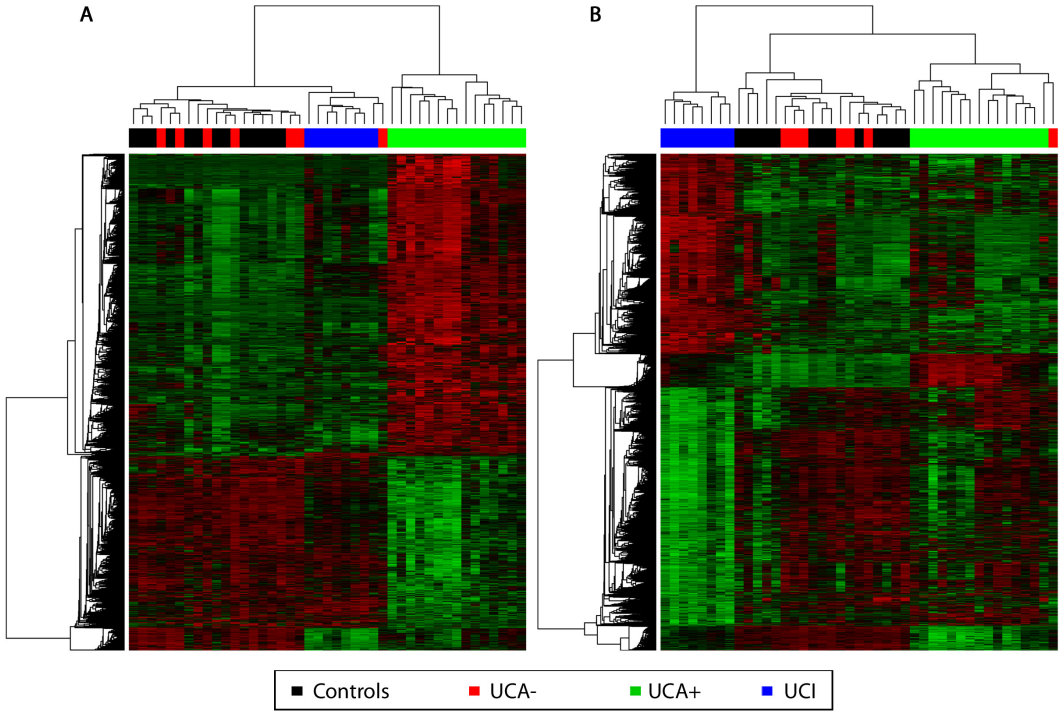

Figura 34: *Heatmap* de l'expressió transcripcional detectada per *microarray*. Cada fila representa una sonda i cada columna una mostra. Nivells elevats d'expressió es mostren en vermell i els nivells baixos en verd. El clúster jeràrquic utilitzant la distància de Pearson i el mètode *average linkage*, s'ha emprat per a la classificació de les mostres i les sondes. Les mostres representades corresponen a: mostres de controls sans (en negre, n=13), mostres sanes de pacients amb colitis ulcerosa activa (en vermell, UCA-, n=7), mostres inflamades de pacients amb colitis ulcerosa activa (en verd, UCA+, n=15) i mostres "curades" de pacients amb colitis ulcerosa en remissió (en blau, UCI, n=8). A) *Heatmap* de les 2.963 sondes diferencialment expressades en les mostres inflamades de pacients amb colitis ulcerosa activa (en verd, UCA+); B) *Heatmap* de les 2.925 sondes diferencialment expressades en les mostres "curades" de pacients amb colitis ulcerosa en remissió (en blau, UCI).

Definint ara el vector que apunta a $(-1,-2)$, i un valor mitjà de correlació $|r| \geq 0.8$ amb una desviació estàndard inferior a la corresponent al primer quartil, identifiquem 728 sondes estretament correlacionades amb el vector (Taula 2).

Gen	Mitjana	Desviació estàndard
1552611_a_at : JAK1	-0.86	0.34
1552673_at : RFX6	0.84	0.15
1552695_a_at : SLC2A13	-0.81	0.34
1552903_at : B4GALNT2	-0.86	0.15
⋮	⋮	⋮
244699_at : AHI1	0.83	0.33
37860_at : ZNF337	0.83	0.36
40189_at : SET	0.87	0.231
57540_at : BRE	-0.80	0.23

Table 2: Resum de les variables més correlacionades amb el vector que apunta a $(-1,-2)$.

A partir de la taxa de canvi (log2-fold change) i el p-valor corregit (FDR) obtinguts de comparar les mostres de mucosa "curada" de pacients amb colitis ulcerosa en remissió respecte les mostres dels controls sans per a les 728 sondes identificades, obtenim que el 95% (694/728) de les variables identificades en el *kernel* PCA són significatives en l'anàlisi d'expressió diferencial (Figura 33, gràfic de la dreta), com és el cas del gen GPR128 (Figura 28).

L'anàlisi d'expressió diferencial ens permet identificar un total de 2.925 sondes corresponents a 2.158 gens significativament alterats en la mucosa "curada" de pacients amb colitis ulcerosa en remissió (Figura 34 B), mentres que l'estudi de correlació de les variables amb la direcció que hem definit només identifica 728 sondes corresponents a 545 gens. Tot i que el nombre de variables identificades pel mètode de correlació és molt menor que l'obtingut utilitzant el mètode d'expressió diferencial, el 86% dels gens identificats en el *kernel* PCA són comuns a l'anàlisi d'expressió diferencial.

Fixant-nos en aquells gens que no concorden entre metodologies, identifiquem alguns gens que podrien ser rellevants, com ADRA2A, detectat en l'anàlisi de correlació del *kernel* PCA (Figura 35). Aquest gen s'ha relacionat amb l'activació

del sistema nerviós simpàtic, la resposta inflamatòria i la severitat de la hipertensió portal [42]. L'expressió disminuida de ADRA2 en la mucosa dels pacients amb colitis ulcerosa en remissió, podria relacionar-se amb un mecanisme antiinflamatori que afavoriria la curació de la mucosa intestinal.

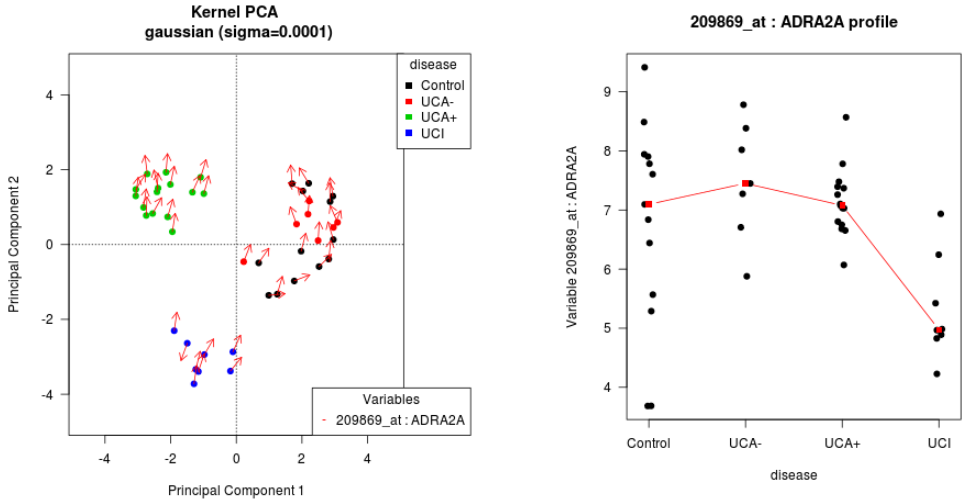

Figura 35: A l'esquerra es mostra el *Kernel* PCA de l'anàlisi transcripcional derivat de mostres de biòpsies intestinals humanes aplicant un kernel gaussià amb $\sigma = 0.0001$ amb representació de variables. Les fletxes en vermell representen el gen ADRA2A. A la dreta es mostra el perfil transcripcional del gen ADRA2A per cada grup de mostres.

Conclusions

La memòria que s'ha realitzat, on es posa en relleu l'ús del *kernel* PCA per a l'anàlisi de dades òmiques, ens permet concloure que:

- Els mètodes basats en funcions *kernel*, per les seves propietats, poden ser una bona aproximació per a la integració de dades, permetent la identificació de relacions no lineals i l'ús de diferents tipus de dades (textos, imatges, números, etc.).
- La metodologia proposada pel departament d'Estadística de la Universitat de Barcelona permet millorar la interpretabilitat de les variables en el *kernel* PCA per un conjunt de dades o per a dades integrades [6].
- La funció desenvolupada en R `KPCAplus` ofereix una extensió de la funció `kpca` de la llibreria `kernlab`, permetent integrar conjunts de dades (sense restricció en el nombre de conjunts) mitjançant *kernels* (de moment s'ha programat pel *kernel* gaussià, polinòmic i ANOVA), representar les variables originals i cercar variables d'interès en el *kernel* PCA.
- La funció `KPCAplus` s'ha programat de forma modular per a poder ampliar facilment les funcions *kernel* aplicables als conjunts de dades (de moment és operatiu per al *kernel* gaussià, polinòmic i ANOVA).
- El disseny de l'aplicació web per a la funció `KPCAplus` permet apropar l'exploració de dades a través del *kernel* PCA a tot tipus d'usuaris, tant els usuaris novells com els experts en R. L'aplicació permet analitzar fins a 3 conjunts de dades proporcionats per l'usuari i exportar els diferents resultats, tant els resultats gràfics com els resultats corresponents a l'anàlisi de cerca de variables. Adicionalment, proporciona dos casos exemple, un amb un conjunt de dades i l'altre amb dos conjunts de dades, il·lustrant l'ús del *kernel* PCA en ambdues situacions.

- S'ha creat la llibreria en R anomenada `KPCAplus`, que conté la funció `KPCAplus` i la funció `KPCAplusGUI` (encarregada d'executar l'aplicació web). La llibreria també inclou un tutorial en HTML al que es pot accedir directament des de l'aplicació web.
- La funció `KPCAplus` es pot utilitzar per a l'anàlisi de dades òmiques.
- Prenent com a referència els resultats d'un estudi transcripcional relacionat amb la colitis ulcerosa, on s'analitzen 22.206 sondes per 43 observacions (biòpsies intestinals humanes), s'ha validat la representació de variables conegudes en el *kernel* PCA (utilitzant un kernel gaussià amb $\sigma = 0.0001$) [34].
- L'estratègia per a la cerca de noves variables en el *kernel* PCA pel conjunt de dades de l'estudi transcripcional ens permet identificar centenars de variables relacionades amb els vectors definits, englobant un gran nombre de les variables significatives en l'anàlisi de dades transcripcionals habitual (LIMMA-*linear models for microarray analysis*). No obstant, el nombre de variables identificades és molt menor que el que s'identifica utilitzant el LIMMA.
- L'ús de la detecció de variables noves en el *kernel* PCA pot afavorir en la detecció de gens rellevants que s'han perdut en l'anàlisi d'expressió diferencial, com en el cas il·lustrat pel gen ADRA2.
- Les mostres biològiques humanes són molt complexes, presentant un ampli ventall de components i interaccions. L'anàlisi i interpretació de resultats d'aquestes demanen una estreta interacció entre els experts de diferentes areas: estadístics, matemàtics, bioinformàtics i biòlegs, entre altres.

Tot i la feina realitzada, encara queda molt camí per recórrer dins del món del *kernel* PCA, la integració de dades i la visualització i cerca de variables. Ens queda pendent avaluar l'ús del *kernel* PCA per a l'anàlisi de més d'un conjunt de dades conegudes. Esperem que amb la progressió dels estudis que estem desenvolupant dins del grup de malaltia inflamatòria intestinal de l'Hospital Clínic i Provincial de Barcelona puguem explorar aquesta aproximació més endavant.

Des d'un punt de vista més metodològic, seria molt interessant investigar els diferents mètodes per a la determinació de les funcions *kernel* i els paràmetres del *kernel* a utilitzar, així com integrar-ho en la funció `KPCAplus` i en l'aplicació web desenvolupada. Adicionalment, manca definir la representació de variables per a un ventall més ampli de funcions *kernels*, com les funcions per a cadenes de caràcters.

Aquestes són algunes de les línies futures de treball més directes, però encara es poden extendre més, avaluant altres mètodes basats en *kernels* i fent un estudi comparatiu entre aquests i el ja estudiat, el *kernel* PCA.

Bibliografia

[1] Andrew R Joyce y Bernhard Ø Palsson. The model organism as a system: integrating'omics' data sets. *Nature Reviews Molecular Cell Biology*, 7(3):198–210, 2006.

[2] Ritchie Marylyn D., Holzinger Emily R., Li Ruowang, Pendergrass Sarah A., y Kim Dokyoon. Methods of integrating data to uncover genotype-phenotype interactions. *Nat Rev Genet*, 16(2):85–97, 2015. ISSN 1471-0056. doi:10.1038/nrg3868;;10.1038/nrg3868.

[3] David Gomez-Cabrero, Imad Abugessaisa, Dieter Maier, Andrew Teschendorff, Matthias Merkenschlager, Andreas Gisel, Esteban Ballestar, Erik Bongcam-Rudloff, Ana Conesa, y Jesper Tegnér. Data integration in the era of omics: current and future challenges. *BMC Systems Biology*, 8(Suppl 2):I1, 2014.

[4] Vaibhav Srivastava, Ogonna Obudulu, Joakim Bygdell, Tommy Löfstedt, Patrik Rydén, Robert Nilsson, Maria Ahnlund, Annika Johansson, Pär Jonsson, Eva Freyhult, et al. OnPLS integration of transcriptomic, proteomic and metabolomic data shows multi-level oxidative stress responses in the cambium of transgenic hipI-superoxide dismutase Populus plants. *BMC genomics*, 14(1):893, 2013.

[5] Chen Meng, Bernhard Kuster, Aedín C Culhane, y Amin Moghaddas Gholami. A multivariate approach to the integration of multi-omics datasets. *BMC bioinformatics*, 15(1):162, 2014.

[6] Ferran Reverter Comes, Esteban Vegas Lozano, y Josep Maria Oller i Sala. Kernel-pca data integration with enhanced interpretability. *BMC Systems Biology, 2014, vol. 8 (S2), num. s6, p. 1-9*, 2014.

[7] Gareth James, Daniela Witten, Trevor Hastie, y Robert Tibshirani. *An introduction to statistical learning*. Springer, 2013.

[8] Ken I Mills, Alexander Kohlmann, P Mickey Williams, Lothar Wieczorek, Wei-min Liu, Rachel Li, Wen Wei, David T Bowen, Helmut Loeffler, Jesus M Hernandez, et al. Microarray-based classifiers and prognosis models identify subgroups with distinct clinical outcomes and high risk of aml transformation of myelodysplastic syndrome. *Blood*, 114(5):1063–1072, 2009.

[9] Bernhard Schölkopf, Koji Tsuda, y Jean-Philippe Vert. *Kernel methods in computational biology*. MIT press, 2004.

[10] Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, págs. 337–404, 1950.

[11] Jean-Philippe Vert, Koji Tsuda, y Bernhard Schölkopf. A primer on kernel methods. *Kernel Methods in Computational Biology*, págs. 35–70, 2004.

[12] John Shawe-Taylor y Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[13] Alexandros Karatzoglou, Alex Smola, Kurt Hornik, y Achim Zeileis. kernlab-an s4 package for kernel methods in r. 2004.

[14] Thomas Hofmann, Bernhard Schölkopf, y Alexander J Smola. Kernel methods in machine learning. *The annals of statistics*, págs. 1171–1220, 2008.

[15] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2009. URL http://www.R-project.org. ISBN 3-900051-07-0.

[16] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2014.

[17] Daniel Peña. *Análisis de datos multivariantes*, tomo 24. McGraw-Hill Madrid, 2002.

[18] Sébastien Lê, Julie Josse, François Husson, et al. Factominer: an r package for multivariate analysis. *Journal of statistical software*, 25(1):1–18, 2008.

[19] Stéphane Dray, Anne-Béatrice Dufour, et al. The ade4 package: implementing the duality diagram for ecologists. *Journal of statistical software*, 22(4):1–20, 2007.

[20] Antoine Lucas. *amap: Another Multidimensional Analysis Package*, 2014. URL http://CRAN.R-project.org/package=amap. R package version 0.8-14.

[21] Unsupervised Learning, Luis Gonzalo Sáncheza, Germán Augusto Osoriob, y Julio Fernando Suárezc. Introducción a kernel acp y otros métodos espectrales aplicados al aprendizaje no supervisado. *Revista Colombiana de Estadística*, 31:19–40, 2008.

[22] Quan Wang. Kernel principal component analysis and its applications in face recognition and active shape models. *arXiv preprint arXiv:1207.3538*, 2012.

[23] Bernhard Schölkopf, Alexander Smola, y Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

[24] Pascal GP Martin, Hervé Guillou, Frédéric Lasserre, Sébastien Déjean, Annaig Lan, Jean-Marc Pascussi, Magali SanCristobal, Philippe Legrand, Philippe Besse, y Thierry Pineau. Novel aspects of pparα-mediated regulation of lipid and xenobiotic metabolism revealed through a nutrigenomic study. *Hepatology*, 45(3):767–777, 2007.

[25] Winston Chang, Joe Cheng, JJ Allaire, Yihui Xie, y Jonathan McPherson. *shiny: Web Application Framework for R*, 2015. URL http://CRAN.R-project.org/package=shiny. R package version 0.11.1.

[26] RStudio. *RStudio: Integrated development environment for R [Computer software]*. RStudio, Boston, MA, 2012. URL http://www.rstudio.org/.

[27] R Core Team. Writing r extensions. *R Foundation for Statistical Computing*, 1999.

[28] C Bernstein, M Fried, JH Krabshuis, H Cohen, R Eliakim, S Fedail, et al. Guías mundiales de la organización mundial de gastroenterología. *Enfermedad inflamatoria intestinal: una perspectiva global*, págs. 1–27, 2009.

[29] Beatriz Sicilia, Raquel Vicente, y Fernando Gomollón. Enfermedad de crohn y colitis ulcerosa: discusión de la epidemiología clásica. *Acta Gastroenterológica Latinoamericana*, 39(2):135–145, 2009.

[30] Kenneth W Schroeder, William J Tremaine, y Duane M Ilstrup. Coated oral 5-aminosalicylic acid therapy for mildly to moderately active ulcerative colitis. *New England Journal of Medicine*, 317(26):1625–1629, 1987.

[31] Ian C Lawrance, Claudio Fiocchi, y Shukti Chakravarti. Ulcerative colitis and crohnâ€™s disease: distinctive gene expression profiles and novel susceptibility candidate genes. *Human Molecular Genetics*, 10(5):445–456, 2001.

[32] BK Dieckgraefe, WF Stenson, JR Korzenik, PE Swanson, y CA Harrington. Analysis of mucosal gene expression in inflammatory bowel disease by parallel oligonucleotide arrays. *Physiological genomics*, 4(1):1–11, 2000.

[33] Christine M Costello, Nancy Mah, Robert Häsler, Philip Rosenstiel, Georg H Waetzig, Andreas Hahn, Tim Lu, Yesim Gurbuz, Susanna Nikolaus, Mario Albrecht, et al. Dissection of the inflammatory bowel disease transcriptome using genome-wide cdna microarrays. *PLoS medicine*, 2(8):e199, 2005.

[34] Núria Planell, Juan J Lozano, Rut Mora-Buch, M Carme Masamunt, Mireya Jimeno, Ingrid Ordás, Miriam Esteller, Elena Ricart, Josep M Piqué, Julián Panés, et al. Transcriptional analysis of the intestinal mucosa of patients with ulcerative colitis in remission reveals lasting epithelial cell alterations. *Gut*, 62(7):967–976, 2013.

[35] Lena Öhman, Hans Törnblom, y Magnus Simrén. Crosstalk at the mucosal border: importance of the gut microenvironment in ibs. *Nature Reviews Gastroenterology & Hepatology*, 12(1):36–49, 2015.

[36] Lance W Peterson y David Artis. Intestinal epithelial cells: regulators of barrier function and immune homeostasis. *Nature Reviews Immunology*, 14(3):141–153, 2014.

[37] Markus F Neurath. Cytokines in inflammatory bowel disease. *Nature Reviews Immunology*, 14(5):329–342, 2014.

[38] Atle van Beelen Granlund, Ann Elisabet Østvik, Øystein Brenna, Sverre H Torp, Bjørn I Gustafsson, y Arne Kristian Sandvik. Reg gene expression in inflamed and healthy colon mucosa explored by in situ hybridisation. *Cell and tissue research*, 352(3):639–646, 2013.

[39] Ying-Yin Ni, Yan Chen, Shun-Yuan Lu, Bi-Ying Sun, Fang Wang, Xiao-Lin Wu, Su-Ying Dang, Guo-Hua Zhang, Hong-Xin Zhang, Yin Kuang, et al. Deletion of gpr128 results in weight loss and increased intestinal contraction frequency. *World journal of gastroenterology: WJG*, 20(2):498, 2014.

[40] Eli Eisenberg y Erez Y Levanon. Human housekeeping genes, revisited. *Trends in Genetics*, 29(10):569–574, 2013.

[41] F Costa, MG Mumolo, L Ceccarelli, M Bellini, MR Romano, C Sterpi, A Ricchiuti, S Marchi, y M Bottai. Calprotectin is a stronger predictive marker of relapse in ulcerative colitis than in crohnâ€™s disease. *Gut*, 54(3):364–368, 2005.

[42] N Shah, M Montes de Oca, D Dhar, M Jover-Cobos, N Alun Davies, R Mookerjee, y R Jalan. P35 treatment with an alpha 2a adrenoreceptor antagonist modulates hepatic inflammation, markedly reduces portal pressure, and improves arterial pressure and hepatic blood flow in cirrhotic rats. *Gut*, 59(Suppl 2):A25–A25, 2010.

Annex

Codi R: funció `KPCAplus`

```
#require packages
require(kernlab)
require(gplots)
require(mixOmics)
require(ggplot2)
require(shiny)


#------------------------------------------>>>> KPCAplus internal functions

derivation=function(x, vars, kernel_function, kparameters, kernelM){
#Comput the direction of maximum growth (derivation at s=0) of determined
    variables for each sample
#Input:
# x: matrix indexed by rows or list of matrixs with row data
# vars: variable names
# kernel_function: kernel function
# kparameters: list with kernel parameters
# kernelM: kernel matrix
#Output: list with the direction of maximum growth (derivation at s=0) of
    determined variables for each sample
 cl <- list()
 for(i in 1:length(vars)){ #for each variable
  if(identical("rbfdot",kernel_function)){ #if gaussian kernel
  Adif <- t(sapply(x[,vars[i]], FUN=function(aa){aa - x[,vars[i]]}))
  res <- -2*unlist(kparameters)*Adif*kernelM #return the derivation for gaussian
       kernel for each data point
  }
  if(identical("polydot",kernel_function)){ #if polinomial kernel
  parms <- unlist(kparameters)
  scaled_x <- parms["scale"]*x
  offscaled_x <- apply(x,1,FUN=function(obs){scaled_x%*%obs})
  res <- parms["degree"]*(offscaled_x+parms["offset"])^(parms["degree"]-1)*x[,
      vars[i]] #return the derivation for polinomial kernel for each data point
  }
  if(identical("anovadot",kernel_function)){ #if ANOVA kernel
  parms <- unlist(kparameters)
  Adif <- t(sapply(x[,vars[i]], FUN=function(aa){x[,vars[i]]-aa}))
```

```
  res <- parms["degree"]*(-1)^(parms["degree"]-1)*parms["sigma"]^(parms["degree"
      ]-1)*Adif^(2*parms["degree"]-2)*exp((-parms["sigma"]*Adif^2)^parms["degree"
      ])
   } #return the derivation for polinomial kernel for each data point
   cl[[i]] <- res #vector with direction of maximum growth of variable 'i' for
       each sample data point
 }
 return(cl)
}


gderivation=function(x, vars, kernel_function, kparameters, kernelM, coef){
#Comput the direction of maximum growth (derivation at s=0) of determined
    variables for each sample
#Input:
# x: matrix indexed by rows or list of matrixs with row data
# vars: variable names
# kernel_function: kernel function
# kparameters: list with kernel parameters
# kernelM: kernel matrix
# coef: linear combination coefficients
#Output: list with the direction of maximum growth (derivation at s=0) of
    determined variables for each sample
 if(is.list(x)){ #when more than 1 dataset is given
  cl <- vector()
  for(j in 1:length(x)){ #for each dataset
  lx <- x[[j]] #matrix of dataset 'j'
  lvars <- vars[vars%in%colnames(lx)] #variables in lx
  lkparameters <- kparameters[[j]] #kernel parameters for dataset 'j'
  lkernelM <- kernelM[[j]] #kernel matrix for dataset 'j'
  lkernel_function <- kernel_function[[j]] #kernel function for dataset 'j'
  if(length(lvars)>0){ #if variable to represent is in dataset 'j'
    cl2 <- derivation(lx, lvars, lkernel_function, lkparameters, lkernelM) #
        Comput the direction of maximum growth (derivation at s=0) of determined
        variables for each sample
    names(cl2) <- lvars #rename each element with the variable name
  }
  if(length(lvars)==0){ #no variable found in dataset 'j'
    cl2 <- NULL #return NULL
  }
  cl <- c(cl,cl2) #list concatenation
  }
 }
 if(is.matrix(x)){ #when 1 dataset is given
  cl <- derivation(x, vars, kernel_function, kparameters, kernelM) #Comput the
      direction of maximum growth (derivation at s=0) of determined variables for
       each sample
  names(cl) <- vars #rename each element with the variable name
 }
 return(cl) #return a list with the direction of maximum growth (derivation at s
     =0) of determined variables for each sample
}
```

```
kernelMatrix_computation=function(KM_kernel_function, KM_kparameters,KM_x){
#Comput the kernel matrix using the kernelMatrix function from krenlab package.
#Input:
# KM_kernel_function: kernel function
# KM_kparameters: list with kernel parameters
# KM_x: data set indexed by row
#Output: Kernel matrix
  if(identical("rbfdot",KM_kernel_function)){ #gaussian kernel
  res <- kernelMatrix(rbfdot(unlist(KM_kparameters)),KM_x)
  } #gaussian formula
  if(identical("polydot",KM_kernel_function)){ #polinomial kernel
  res <- kernelMatrix(polydot(unlist(KM_kparameters)),KM_x)
  } #polynomial kernel

  if(identical("anovadot",KM_kernel_function)){ #ANOVA kernel
  res <- kernelMatrix(anovadot(unlist(KM_kparameters)),KM_x)
  } #base radial ANOVA formula
  return(res)
}



#------------------------------------------>>>> KPCAplus function

KPCAplus = function(x, kernel_function, kparameters, xpc=1, ypc=2, factors=NULL,
     variables=NULL, combination="+",  var_combination=NULL, coef_combination=
    NULL, scale=0.5, plot.kPCA=TRUE, plot.profile=FALSE, variables_discovery=NULL
    , discovery.col="orange", weights=NULL, text.sample=FALSE, show.legend=TRUE,
    ...){
#... -> refers to other kpca plot parameters


############################################
# Parameters control                       #
############################################

if(!is.matrix(x) & !is.list(x)) stop("'x' must be a matrix indexed by row\n")
if(!is.character(kernel_function) & !is.list(kernel_function)) stop("'kernel_
    function' must be a character or a list of characters\n")
if(!is.list(kparameters)) stop("'kparameters' must be a list\n")
if(!is.numeric(xpc) | xpc>8 | xpc<=0) stop("'xpc' must be numeric, range 1-8.\n"
    )
if(!is.numeric(ypc) | ypc>8 | xpc<=0) stop("'ypc' must be numeric, range 1-8.\n"
    )
if(!is.null(factors) & !is.data.frame(factors)) stop("'factors' must be a
    dataframe\n")
if(!is.null(factors)) factors <- data.frame(apply(factors,2,FUN=function(ff){
    return(as.factor(ff))}))
if(!is.null(variables) & !is.character(variables)) stop("'variables' must be a
    character vector\n")
if(!is.character(combination)) stop("'combination' must be a character defining
    an operation: '+', '*'\n")
if(!is.null(var_combination) & !is.character(var_combination)) stop("'var_
    combination' must be a character vector\n")
```

```
if(!is.null(coef_combination) & !is.numeric(coef_combination)) stop("'coef_
    combination' must be a numeric vector\n")
if(!is.numeric(scale)) stop("'scale' must be numeric\n")
if(!is.logical(plot.kPCA)) stop("'plot.kPCA' must be logical\n")
if(!is.logical(plot.profile)) stop("'plot.profile' must be logical\n")
if(plot.profile & !is.null(variables) & is.null(factors)) stop("'plot.profile'
    must be FALSE or 'factors' should be provided \n")
if(!is.null(variables_discovery) & !is.numeric(variables_discovery)) stop("'
    variables_discovery' must be a numeric vector\n")
if(!is.null(variables) | !is.null(var_combination)){
 all_var <- c(variables, var_combination)
 if(is.list(x)) data_var <- unlist(lapply(x,FUN=function(xx){return(colnames(xx)
     )}))
 if(is.matrix(x)) data_var <- colnames(x)
 if(sum(all_var%in%data_var)!=length(all_var)) stop("'variables' or 'var_
     combination' sepcified are not in the data set provided\n")
 if(!is.null(weights) & !is.numeric(weights)) stop("'weight' must be numeric\n")
 if(is.numeric(weights) & !is.list(x)) stop("'weight' defined when only one data
      set used\n")
 if(is.numeric(weights) & length(x)!=length(weights))stop("'weight' must have
     the same lenght as the number of datasets analyzed\n")
 if(!is.character(combination)) stop("'combination' must be character. '
     combination' defines the combination variables equation operation, being
     addition '+', multiplicative '*', substraction '-' or division '\\'\n")
}
if(!is.logical(show.legend)) stop("'show.legend' must be logical\n")

kpca_plus_results <- list() #object with resulta to return


##############################################
# Compute KPCA                               #
##############################################

xpc <- as.integer(xpc) #integer
ypc <- as.integer(ypc) #integer
if(is.list(x)){ #when more than 1 dataset is given
 nd <- length(x) #number of datasets
 if(is.null(weights)) weights <- rep(1,length(x)) #weights of datasets
  kernelM <- list() #list to save the kernel matrix for each dataset

  for(i in 1:length(x)){  #compute kernel matrix for each dataset
    kernelM[[i]] <- kernelMatrix_computation(KM_kernel_function=kernel_function
        [[i]], KM_kparameters=kparameters[[i]], KM_x=x[[i]])*weights[i]
    kernel_matrix <- as.kernelMatrix(Reduce("+", kernelM)) #kernel matrix of
        combinated kernels (addition)
  }
  kpca_result <- kpca(as.kernelMatrix(Reduce("+", kernelM)), features=8) #kpca
      from combinated kernels (kernlab)
}
if(is.matrix(x)){ #when 1 dataset is given
 nd <- 1 #number of datasets
 kpca_result <- kpca(x, kernel=kernel_function, kpar=kparameters, features=8,
     ...) #kpca from given data (kernlab)
```

```
 kernelM <- kernelMatrix_computation(KM_kernel_function=kernel_function, KM_
     kparameters=kparameters, KM_x=x) #compute kernel matrix for dataset
 kernel_matrix <- kernelM #kernel matrix
}
kpca_plus_results$datasets <- list(number_data_sets=nd, data=x) #list with numer
     of datasets and raw datasets
kpca_plus_results$KPCA <- list(weights=weights, kernel=kernel_function, kernel_
    parameters=kparameters, kernel_matrix=kernel_matrix, kpca=kpca_result) #list
    with weights, kernel function, kernel parameters, kernel matrix and output of
     kpca.
if(plot.kPCA==TRUE){ #visualizing the KPCA in a biplot
par()
if(show.legend==TRUE) par(mar=c(5.1, 4.1, 4.1, 8.1), xpd=TRUE)
limi <- max(abs(min(rotated(kpca_result)[,xpc],rotated(kpca_result)[,ypc])), max
    (rotated(kpca_result)[,xpc],rotated(kpca_result)[,ypc]))
inferior_lim <- (-1*limi) - 1
superior_lim <- limi + 1
if(!is.null(factors)){ #factors given
 col_kpca <- as.numeric(factors[,1]) #assign colors to first factor
 pch_kpca <- 19
 if(ncol(factors)>1){ #more than 1 factor given
  pch_kpca <- as.numeric(factors[,2])-1 #assign points symbols to second factor
 }
 plot(rotated(kpca_result)[,c(xpc,ypc)], col=col_kpca, pch=pch_kpca, xlab=paste(
     "Principal Component", xpc, sep=" "), ylab=paste("Principal Component", ypc,
      sep=" "), las=1, xlim=c(inferior_lim, superior_lim), ylim=c(inferior_lim,
     superior_lim), ...) #biplot KPCA
 if(show.legend==TRUE){
  legend("topright", inset=c(-0.2, 0), legend=levels(factors[,1]), pch=15, col
      =1:length(levels(factors[,1])), title=names(factors)[1]) #legend first
      factor
  if(ncol(factors)>1){ #more than 1 factor given
   legend("right", inset=c(-0.2,0), legend=levels(factors[,2]), pch=0:(length(
       levels(factors[,2]))-1), title=names(factors)[2]) #legend second factor
  }
 }
}
if(is.null(factors)){ #no factors given
 col_kpca <- "black" #color points
 pch_kpca <- 19 #symbol points
 plot(rotated(kpca_result)[,c(xpc,ypc)], col=col_kpca, pch=pch_kpca, xlab=paste(
     "Principal Component", xpc, sep=" "), ylab=paste("Principal Component", ypc,
      sep=" "), las=1, xlim=c(inferior_lim, superior_lim), ylim=c(inferior_lim,
     superior_lim), ...) #biplot KPCA
}
if(text.sample==TRUE){ #if TRUE print sample names
 if(is.list(x)) snames <- rownames(x[[1]]) #sample names when more than 1
     dataset is given
 if(is.matrix(x)) snames <- rownames(x) #sample names when 1 dataset is given
 text(rotated(kpca_result)[,c(xpc,ypc)], labels=snames, cex=0.6, adj=1) #print
     sample names to KPCA biplot
}
abline(v=0, lty=3, xpd=FALSE) #add a vertical dotted line at 0
abline(h=0, lty=3, xpd=FALSE) #add an horitzontal dotted line at 0
```

```
}


##############################################
# Represent original variables              #
##############################################

#origin point for each sample in the kpca representation
x0 <- rotated(kpca_result)[,xpc] #projection points of PC x-axis
y0 <- rotated(kpca_result)[,ypc] #projection points of PC y-axis
Vcuc <- pcv(kpca_result)[,c(xpc,ypc)] #matrix with principal components
MM <- (diag(nrow=length(x0))) - (matrix(1/length(x0), ncol=length(x0), nrow=
    length(x0))) #part of the KPCA images projection formula

#-----> Linear combination of original variables representation

if(!is.null(var_combination)){ # variables for combination defined
 if(length(var_combination)==1) stop("'var_combination' must be a character
     vector of length >1\n") #we need more than one variable defined in order to
     do de variable combination
 if(!is.null(coef_combination) & length(coef_combination)!=length(var_
     combination)) stop("'var_combination' and 'coef_combination' must have the
     same length\n") #verify that the number of coeficients, if defined, are the
     same as number of variables
 if(is.null(coef_combination)) coef_combination <- rep(1, length(var_combination
     ))
 cl <- gderivation(x, var_combination, kernel_function, kparameters, kernelM) #
     kernel derivation as part of the KPCA images projection formula
 for(vc in 1:length(cl))
  cl[[vc]] <- cl[[vc]]*coef_combination
 res <- Reduce(combination, cl) #linear combination
 x1 <- res %*% MM %*% Vcuc #images projection in KPCA
 norm_x1 <- apply(x1,1,FUN=function(g){sqrt(sum(g^2))}) #variable standarization
 x0n <- (x1/norm_x1)*scale #reduce arrow length
 if(plot.kPCA==TRUE) arrows(x0, y0, x1=(x0+x0n[,1]), y1=(y0+x0n[,2]), col="black
     ", length=0.10, lty=1, lwd=1) #plot the arrow pointing the direction of
     maximum growth
 comvar <- var_combination[1] #variables for legend
 for(i in 2:length(var_combination)){ #if more than one variable, define legend
 comvar <- paste(comvar, var_combination[i], sep=combination) #variable legend
 }
 if(is.null(variables) & plot.kPCA==TRUE & show.legend==TRUE) legend("
     bottomright", inset=c(-0.2,0), legend=comvar, pch=c("-"),col="black",title="
     Variables", bg="white") #print variable combination legend
}

#-----> Original variables representation

if(!is.null(variables)){ # variables defined
 cl <- gderivation(x, variables, kernel_function, kparameters, kernelM) #kernel
     derivation as part of the KPCA images projection formula
 for(i in 1:length(variables)){ #for each variable
  res <- cl[[i]] #get the maximum growth for variable 'i' for each sample
  x1 <- res %*% MM %*% Vcuc #images projection in KPCA
```

```
 norm_x1 <- apply(x1,1,FUN=function(g){sqrt(sum(g^2))}) #variable
     standarization
 x0n <- (x1/norm_x1)*scale #reduce arrow length
  if(plot.kPCA==TRUE) arrows(x0, y0, x1=(x0+x0n[,1]), y1=(y0+x0n[,2]), col=i+1,
       length=0.10, lty=1, lwd=1) #plot the arrow pointing the direction of
      maximum growth
}
if(!is.null(var_combination) & plot.kPCA==TRUE & show.legend==TRUE) legend("
    bottomright", inset=c(-0.2,0), legend=c(comvar,names(cl)), pch=c("-"),col=c
    (1:(length(variables)+1)),title="Variables", bg="white") #plot legend with
    variable combination and individual variables
if(is.null(var_combination) & plot.kPCA==TRUE & show.legend==TRUE) legend("
    bottomright", inset=c(-0.2,0), legend=names(cl), pch=c("-"),col=c(2:(length(
    variables)+1)),title="Variables", bg="white") #plot legend with individual
    variables

 #plot variable profile
if(plot.profile){
par(mar=c(6.1, 4.1, 4.1, 8.1), xpd=TRUE)
 if(is.list(x)){
  for(k in 1:length(x)){
 xx <- x[[k]]
 vars <- variables[variables%in%colnames(xx)]
 if(length(vars)>0){
  for(i in 1:length(vars)){
    if(ncol(factors)==1){
     stripchart(xx[,vars[i]] ~ factors[,1], method = "jitter", jitter=0.05,
         main=paste(vars[i],"profile", sep=" "), col="black", vertical= TRUE,
         pch=19, ylab=paste("Variable",variables[i],sep=" "), xlab=names(factors
         )[1], las=1)
   points(1:length((levels(factors[,1]))), tapply(xx[,vars[i]],factors[,1],
       median), col="red", type="b", pch=15)
      }
    if(ncol(factors)>1){
     stripchart(xx[,vars[i]] ~ factors[,1], method = "jitter", jitter=0.05,
         main=paste(vars[i],"profile", sep=" "), col=as.numeric(factors[,2]),
         vertical= TRUE, pch=NA, ylab=paste("Variable",vars[i],sep=" "), xlab=
         names(factors)[1], las=1)
   points(jitter(as.numeric(factors[,1]),0.2), xx[,variables[i]], col=as.
       numeric(factors[,2]), pch=19)
   for(j in 1:length(levels(factors[,2]))){
     sel <- which(factors[,2]==levels(factors[,2])[j])
     points(1:length((levels(factors[,1]))), tapply(xx[sel,vars[i]],factors[sel
         ,1],median), col=j, type="b", pch=15)
   }
   legend("topright", inset=c(-0.2,0), legend=levels(factors[,2]), pch=19, col
       =1:length(levels(factors[,2])), title=names(factors)[2])
    }
  }
 }
  }
 }
 if(is.matrix(x)){
  for(i in 1:length(variables)){
```

```
    if(ncol(factors)==1){
    stripchart(x[,variables[i]] ~ factors[,1], method = "jitter", jitter=0.05,
        main=paste(variables[i],"profile", sep=" "), col="black", vertical= TRUE,
         pch=19, ylab=paste("Variable",variables[i],sep=" "), xlab=names(factors)
        [1], las=1)
  points(1:length((levels(factors[,1]))), tapply(x[,variables[i]],factors[,1],
      median), col="red", type="b", pch=15)
    }
    if(ncol(factors)>1){
    stripchart(x[,variables[i]] ~ factors[,1], method = "jitter", jitter=0.05,
        main=paste(variables[i],"profile", sep=" "), col=as.numeric(factors[,2]),
         vertical= TRUE, pch=NA, ylab=paste("Variable",variables[i],sep=" "),
        xlab=names(factors)[1], las=1)
  points(jitter(as.numeric(factors[,1]),0.2), x[,variables[i]], col=as.numeric(
      factors[,2]), pch=19)
  for(j in 1:length(levels(factors[,2]))){
    sel <- which(factors[,2]==levels(factors[,2])[j])
    points(1:length((levels(factors[,1]))), tapply(x[sel,variables[i]],factors[
        sel,1],median), col=j, type="b", pch=15)
  }
  legend("topright", inset=c(-0.2,0), legend=levels(factors[,2]), pch=19, col=1:
      length(levels(factors[,2])), title=names(factors)[2])
    }
   }
  }
 }
}


##############################################
# Variables discovery                        #
##############################################

if(!is.null(variables_discovery)){ #done variable discovery
if(plot.kPCA==TRUE){
 arrows(x0=variables_discovery[1], y0=variables_discovery[2], x1=variables_
     discovery[3], y1=variables_discovery[4], col=discovery.col, length=0.10, lty
     =1, lwd=3)} #print the arrow showing the direction of interest in the KPCA
 if(is.matrix(x)){ #when one dataset transform all the input objects as list.
  x <- list(x)
  kernel_function <- list(kernel_function)
  kparameters <- list(kparameters)
  kernelM <- list(kernelM)
 }
 cl2 <- vector()
 vars <- vector()
 dataset <- vector()
 for(k in 1:length(x)){ #for each dataset
  variables <- colnames(x[[k]]) #variables of dataset 'k'
  cl <- gderivation(x[[k]], variables, kernel_function[[k]], kparameters[[k]],
      kernelM[[k]]) #kernel derivation as part of the KPCA images projection
      formula. Give the direction of maximum growth for each variable in each
      sample
  scl2 <- list()
```

```
  for(i in 1:length(variables)){ #for each variable
   res <- cl[[i]] #derivation variable 'i'
   x1 <- res %*% MM %*% Vcuc #images projection in KPCA for varaible 'i'
   scl2[[i]] <- x1
  }
  cl2 <- c(cl2,scl2) #points
  vars <- c(vars,variables) #variables
  dataset <- c(dataset,rep(k,length(variables))) #dataset
 }
 norm_discovery <- variables_discovery[3:4]/sqrt(sum(variables_discovery[3:4]^2)
     )
 alignement <-lapply(cl2,FUN=function(dd){
  if(is.matrix(dd)){ rdd <- dd%*%norm_discovery / apply(dd,1,FUN=function(g){
      sqrt(sum(g^2))})}
  else { rdd <- dd*norm_discovery / sqrt(sum(dd^2))}
  return(rdd)})
 sd.l    <- unlist(lapply(alignement,sd))
 mean.l <-unlist(lapply(alignement,mean))
 alignement_result <- data.frame(names = vars, dataset = dataset, alin.mean =
     mean.l, alin.sd = sd.l)
 kpca_plus_results$variable_discovery <- alignement_result
}
return(kpca_plus_results)
} #end function


#------------------------------------------>>>> KPCAplus shiny

KPCAplusGUI=function(){
shiny::runApp(system.file("appdir", package="KPCAplus"))
}
```