

## INDICE DE ANEXOS

Anexo 1:	Algoritmo del programa DetecciónBK	1
Anexo 2:	Algoritmo del programa AnalizaDF	4



### **Anexo 1: Programa Detección BK**

```
close all; clear all; clc  
tic;
```

```

direc={'bacilos\' 'nobacilosa\' 'nobacilosb\'};

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%PROCEDIMIENTO BACILOS
A%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
muestra=350;          %número de objetos
k=1;                 %bacilos
fuente=direc{k};
carpeta=strcat('D:\Javier\Tesis\',fuente);

for t=1:muestra
file=strcat(carpeta,'objeto',num2str(t));
load (file, 's');    %en s guardo las coordenadas, "x" en
                    %una columna "y" en otra columna
l=length(s);        %cantidad de coordenadas (número de
pixeles)
s=imresize(s,[317 2]); %Cambio de tamaño de imagen
s=s(:,1)+i*s(:,2);  %s=sx+isy (transformación a número
complejo)
z=fft(s);           %Obtención de descriptores de Fourier
z(1)=0;             %Invariante a traslación
z=z/abs(z(317));    %Invariante a escala (Se dividen los
                    %descriptores entre la magnitud del
                    %2do descriptor (317)
magnitudes(t,:)=abs(z); %Matriz final con magnitudes de
                    %descriptores de bacilos
end
clear t l s muestra k z

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%PROCEDIMIENTO PARA NO BACILOS A
A%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
muestra=72;          %número de objetos
k=2;                 %no bacilos A
fuente=direc{k};
carpeta=strcat('D:\Javier\Tesis\',fuente);

for t=1:muestra
file=strcat(carpeta,'objeto',num2str(t));
load (file, 's');    %en s guardo las coordenadas, "x" en
                    %una columna "y" en otra columna
l=length(s);
s=imresize(s,[317 2]); %Cambio de tamaño de imagen
s=s(:,1)+i*s(:,2);
z=fft(s);
z(1)=0;
z=z/abs(z(317));
magnitudesNBA(t,:)=abs(z); %Matriz final con magnitudes de
                    %descriptores de no bacilos A
end
clear t l s muestra k z

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%PROCEDIMIENTO PARA NO BACILOS B
A%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
muestra=58;          %número de objetos
k=3;                 %no bacilos B

```

```

fuente=direc{k};
carpeta=strcat('D:\Javier\Tesis\',fuente);
for t=1:muestra
file=strcat(carpeta,'objeto',num2str(t));
load (file, 's');           %en s guardo las coordenadas, "x" en
                             una columna "y" en otra columna

l=length(s);
s=imresize(s,[317 2]);     %Cambio de tamaño de imagen
s=s(:,1)+i*s(:,2);
z=fft(s);
z(1)=0;
z=z/abs(z(317));
magnitudesNBB(t,:)=abs(z); %Matriz final con magnitudes de
                             descriptores de no bacilos B

end
clear t l s muestra k z

universo=[magnitudes;magnitudesNBA;magnitudesNBB];
clear magnitudes magnitudesNBA magnitudesNBB

n=0;                         % n indica elementos válidos en el
vector

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%SUBOPTIMAL FEATURE
SELECTION%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Filtro=[2:316];
m=length(Filtro);
x=14;                         %cantidad de descriptores a obtener
for h=1:m-x;
b=length(Filtro);
DFA(1,:)=Filtro(1:b-1)]; %DFA matriz que tendrá combinaciones
for i=1:b
if i<b
DFA(i+1,:)=Filtro(:,1:b-(i+1)) Filtro(:,b-i+1:b)];
%Completando matriz con combinaciones
end
end
%se completa la matriz DFA que
contiene todas las combinaciones

for j=1:b
for k=1:3
%se obtiene el resultado de acuerdo a
DFA en cada muestra
NS=k;                         %Indica la muestra que se quiere
evaluar NS=1,2 ó 3
[PA,SS,ST,err]=analizaDF(DFA(j,:),universo,NS);
%Se clasifica
RESULTADOS(k,1:4)=[PA,SS,ST,err];
clear PA SS ST err
end

%El criterio para escoger es el promedio entre 3 sample
considerando el Porcentaje de acierto

C(j,:)=mean(RESULTADOS(:,1));

%Adicionalmente obtengo en D el promedio entre 3 sample
considerando la Sensibilidad

```

```

D(j,:) = mean(RESULTADOS(:,2));

%Adicionalmente obtengo en E el promedio entre 3 sample
considerando la Especificidad

E(j,:) = mean(RESULTADOS(:,3));

clear NS k m n i
end

posc=find(C==max(C));
posc=max(posc);

Filtro=[DFA(posc,:)]; %Nuevo filtro. Aquí van quedando las
combinaciones con mejor promedio de
acuerdo al criterio (acierto,
sensibilidad o especificidad)

if length(Filtro)==x;
    DFClasificadores=Filtro;
    ACF=C(posc);
    SSF=D(posc);
    STF=E(posc);
end
clear DFA
clear C D E
end
clear h p j posc x b RESULTADOS
tiempo=toc/60; %minutos que toma correr el programa
clear z

```

## Anexo 2: Programa AnalizaDF

```

function[PA,SS,ST,err]=AnalizaDF(DFA,universo,NS)
%Función que recibe la muestra total, el o los descriptores que
se desean
%analizar y el sample con que se quiere realizar el test.
%Como resultado se obtiene la sensibilidad, especificidad y el
porcentaje
%de acierto aplicando LDA como clasificador

universo=[universo(:,DFA)]; %Me quedo solo con los descriptores
seleccionados

if NS==1
sample=[universo(1:116,:);universo(351:374,:);universo(423:441,:)]
]; real=[repmat(1,116,1);repmat(2,24,1);repmat(3,19,1)];

```

```

training=[universo(117:350,:);universo(375:422,:);universo(442:48
0,:)];
group=[repmat(1,234,1);repmat(2,48,1);repmat(3,39,1)];
end
if NS==2
sample=[universo(117:233,:);universo(375:398,:);universo(442:460,
:)]];
real=[repmat(1,117,1);repmat(2,24,1);repmat(3,19,1)];
training=[universo(1:116,:);universo(234:374,:);universo(399:441,
:);universo(461:480,:)]];
group=[repmat(1,233,1);repmat(2,48,1);repmat(3,39,1)];
end
if NS==3
sample=[universo(234:350,:);universo(399:422,:);universo(461:480,
:)]];
real=[repmat(1,117,1);repmat(2,24,1);repmat(3,20,1)]; %159x1
training=[universo(1:233,:);universo(351:398,:);universo(423:460,
:)]];
group=[repmat(1,233,1);repmat(2,48,1);repmat(3,38,1)];
end

[class,err]=classify(sample,training,group);
RESTA=class-real;
PA=0;
VP=0;
VN=0;
FN=0;
FP=0;

for i=1:length(RESTA)
if RESTA(i)== 0;
    PA=PA+1; %PA son los aciertos
    if real(i)==1
        VP=VP+1;
    else
        VN=VN+1;
    end
end
if RESTA(i)== 1;
    if real(i)== 1
        FN=FN+1;
    else
        VN=VN+1;
    end
end
if RESTA(i)==2;
    FN=FN+1;
end
if RESTA(i)==-2;
    FP=FP+1;
end
if RESTA(i)==-1;
    if real(i)==2;
        FP=FP+1;
    else
        VN=VN+1;
    end
end
end

```

```
end
[m,n]=size(sample);
PA=PA*100/m;           %Porcentaje de acierto
POSITIVO=VP+FN;
NEGATIVO=FP+VN;
TOTAL=VP+VN+FP+FN;
SS=VP/(VP+FN);       %SenSibility
ST=VN/(FP+VN);       %SpecificiTy
```



## Anexo 1: Programa DetecciónBK

```
close all; clear all; clc
tic;
direc={'bacilos\' 'nobacilosa\' 'nobacilosb\'};

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PROCEDIMIENTO BACILOS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
muestra=350;          %número de objetos
k=1;                 %bacilos
fuente=direc{k};
carpeta=strcat('D:\Javier\Tesis\',fuente);

for t=1:muestra
file=strcat(carpeta,'objeto',num2str(t));
load (file, 's');          %en s guardo las coordenadas, "x" en
                          %una columna "y" en otra columna
l=length(s);             %cantidad de coordenadas (número de
                          %píxeles)
s=imresize(s,[317 2]);   %Cambio de tamaño de imagen
s=s(:,1)+i*s(:,2);      %s=sx+isy (transformación a número
                          %complejo)
z=fft(s);                %Obtención de descriptores de Fourier
z(1)=0;                  %Invariante a traslación
z=z/abs(z(317));         %Invariante a escala (Se dividen los
                          %descriptores entre la magnitud del
                          %2do descriptor (317))
magnitudes(t,:)=abs(z);  %Matriz final con magnitudes de
                          %descriptores de bacilos
end
clear t l s muestra k z

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PROCEDIMIENTO PARA NO BACILOS A %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
muestra=72;          %número de objetos
k=2;                 %no bacilos A
fuente=direc{k};
carpeta=strcat('D:\Javier\Tesis\',fuente);

for t=1:muestra
file=strcat(carpeta,'objeto',num2str(t));
load (file, 's');          %en s guardo las coordenadas, "x" en
                          %una columna "y" en otra columna
l=length(s);
s=imresize(s,[317 2]);   %Cambio de tamaño de imagen
s=s(:,1)+i*s(:,2);
z=fft(s);
z(1)=0;
z=z/abs(z(317));
magnitudesNBA(t,:)=abs(z); %Matriz final con magnitudes de
                          %descriptores de no bacilos A
end
clear t l s muestra k z
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PROCEDIMIENTO PARA NO BACILOS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
muestra=58; %número de objetos
k=3; %no bacilos B
fuente=direc{k};
carpeta=strcat('D:\Javier\Tesis\',fuente);
for t=1:muestra
file=strcat(carpeta,'objeto',num2str(t));
load (file, 's'); %en s guardo las coordenadas, "x" en
%una columna "y" en otra columna

l=length(s);
s=imresize(s,[317 2]); %Cambio de tamaño de imagen
s=s(:,1)+i*s(:,2);
z=fft(s);
z(1)=0;
z=z/abs(z(317));
magnitudesNBB(t,:)=abs(z); %Matriz final con magnitudes de
%descriptores de no bacilos B

end
clear t l s muestra k z

universo=[magnitudes;magnitudesNBA;magnitudesNBB];
clear magnitudes magnitudesNBA magnitudesNBB

n=0; % n indica elementos válidos en el
%vector

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SUBOPTIMAL FEATURE SELECTION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Filtro=[2:316];
m=length(Filtro);
x=14; %cantidad de descriptores a obtener
for h=1:m-x;
b=length(Filtro);
DFA(1,:)=Filtro(1:b-1)]; %DFA matriz que tendrá combinaciones
for i=1:b
if i<b
DFA(i+1,:)=Filtro(:,1:b-(i+1)) Filtro(:,b-i+1:b)];
%Completando matriz con combinaciones
end
end %se completa la matriz DFA que
%contiene todas las combinaciones

for j=1:b
for k=1:3 %se obtiene el resultado de acuerdo a
%DFA en cada muestra
NS=k; %Indica la muestra que se quiere
%evaluar NS=1,2 ó 3
[PA,SS,ST,err]=AnalizaDF(DFA(j,:),universo,NS);
%Se clasifica
RESULTADOS(k,1:4)=[PA,SS,ST,err];
clear PA SS ST err
end

%El criterio para escoger es el promedio entre los 3 samples
considerando el Porcentaje de acierto

C(j,:)=mean(RESULTADOS(:,1));

```



```

%Adicionalmente obtengo en D el promedio entre los 3 samples
considerando la Sensibilidad

D(j,:) = mean(RESULTADOS(:,2));

%Adicionalmente obtengo en E el promedio entre los 3 samples
considerando la Especificidad

E(j,:) = mean(RESULTADOS(:,3));

clear NS k m n i
end

posc=find(C==max(C));      %selecciono de acuerdo al porcentaje
posc=max(posc);           %de acierto

Filtro=[DFA(posc,:)];     %Nuevo filtro. Aquí van quedando las
                           %combinaciones con mejor promedio de
                           %acuerdo al criterio (acierto,
                           %sensibilidad o especificidad)

if length(Filtro)==x;
    DFClasificadores=Filtro;
    ACF=C(posc);
    SSF=D(posc);
    STF=E(posc);
end
clear DFA
clear C D E
end
clear h p j posc x b RESULTADOS
tiempo=toc/60;            %minutos que toma la ejecución del
                           %programa
clear z

```

## Anexo 2: Programa AnalizaDF

```
function[PA,SS,ST,err]=AnalizaDF(DFA,universo,NS)
%Función que recibe la muestra total, el o los descriptores que
%se desean analizar y el sample con que se quiere realizar el
%test. Como resultado se obtiene la sensibilidad, especificidad y
%el porcentaje de acierto, aplicando LDA como clasificador

universo=[universo(:,DFA)]; %Me quedo solo con los descriptores
seleccionados

if NS==1
sample=[universo(1:116,:);universo(351:374,:);universo(423:441,:)
];
real=[repmat(1,116,1);repmat(2,24,1);repmat(3,19,1)];
training=[universo(117:350,:);universo(375:422,:);universo(442:48
0,:)];
group=[repmat(1,234,1);repmat(2,48,1);repmat(3,39,1)];
end

if NS==2
sample=[universo(117:233,:);universo(375:398,:);universo(442:460,
:)];
real=[repmat(1,117,1);repmat(2,24,1);repmat(3,19,1)];
training=[universo(1:116,:);universo(234:374,:);universo(399:441,
:);universo(461:480,:)];
group=[repmat(1,233,1);repmat(2,48,1);repmat(3,39,1)];
end

if NS==3
sample=[universo(234:350,:);universo(399:422,:);universo(461:480,
:)];
real=[repmat(1,117,1);repmat(2,24,1);repmat(3,20,1)]; %159x1
training=[universo(1:233,:);universo(351:398,:);universo(423:460,
:)];
group=[repmat(1,233,1);repmat(2,48,1);repmat(3,38,1)];
end

[class,err]=classify(sample,training,group);
RESTA=class-real;
PA=0;
VP=0;
VN=0;
FN=0;
FP=0;

for i=1:length(RESTA)
if RESTA(i)== 0;
    PA=PA+1; %PA son los aciertos
    if real(i)==1
        VP=VP+1;
    else
        VN=VN+1;
    end
end
if RESTA(i)== 1;
    if real(i)== 1
        FN=FN+1;
    else
```

```

        VN=VN+1;
    end
end
if RESTA(i)==2;
    FN=FN+1;
end
if RESTA(i)==-2;
    FP=FP+1;
end
if RESTA(i)==-1;
    if real(i)==2;
        FP=FP+1;
    else
        VN=VN+1;
    end
end
end
end
[m,n]=size(sample);
PA=PA*100/m; %Porcentaje de acierto
POSITIVO=VP+FN;
NEGATIVO=FP+VN;
TOTAL=VP+VN+FP+FN;
SS=VP/(VP+FN); %SenSibility
ST=VN/(FP+VN); %SpecificiTy

```

