Project 11 NMF and multi-omics data integration

Team K - Minghang Li

May 25, 2023

Problem 31: Uniqueness of NMF solutions

The solution is not necessarily unique. The easiest way is to give couter example.

Suppose we have:

$$V = \begin{bmatrix} 5 & 10 \end{bmatrix}$$

Then it is possible to decompose it into given $k = 1$:

$$H = \begin{bmatrix} 1 & 2 \end{bmatrix}, \quad W = \begin{bmatrix} 5 \end{bmatrix}$$

However, suppose we have $\tilde{W} = [2]$, we can also find $\tilde{H}$ such that:

$$\tilde{H} = \tilde{W}^{-1} W H = \begin{bmatrix} 5/2 & 5 \end{bmatrix}$$

So that $WH = \tilde{W}\tilde{H}$ but $W \neq \tilde{W}$ and $H \neq \tilde{H}$. In general if we have square $\tilde{W}$ and it is inversible the solution is not unique.

(Of course I'm not satisfied with this proof because it only proves the non-uniqueness when $K = M$. I'm not sure how to tackle this problem in a more general scenario though…)

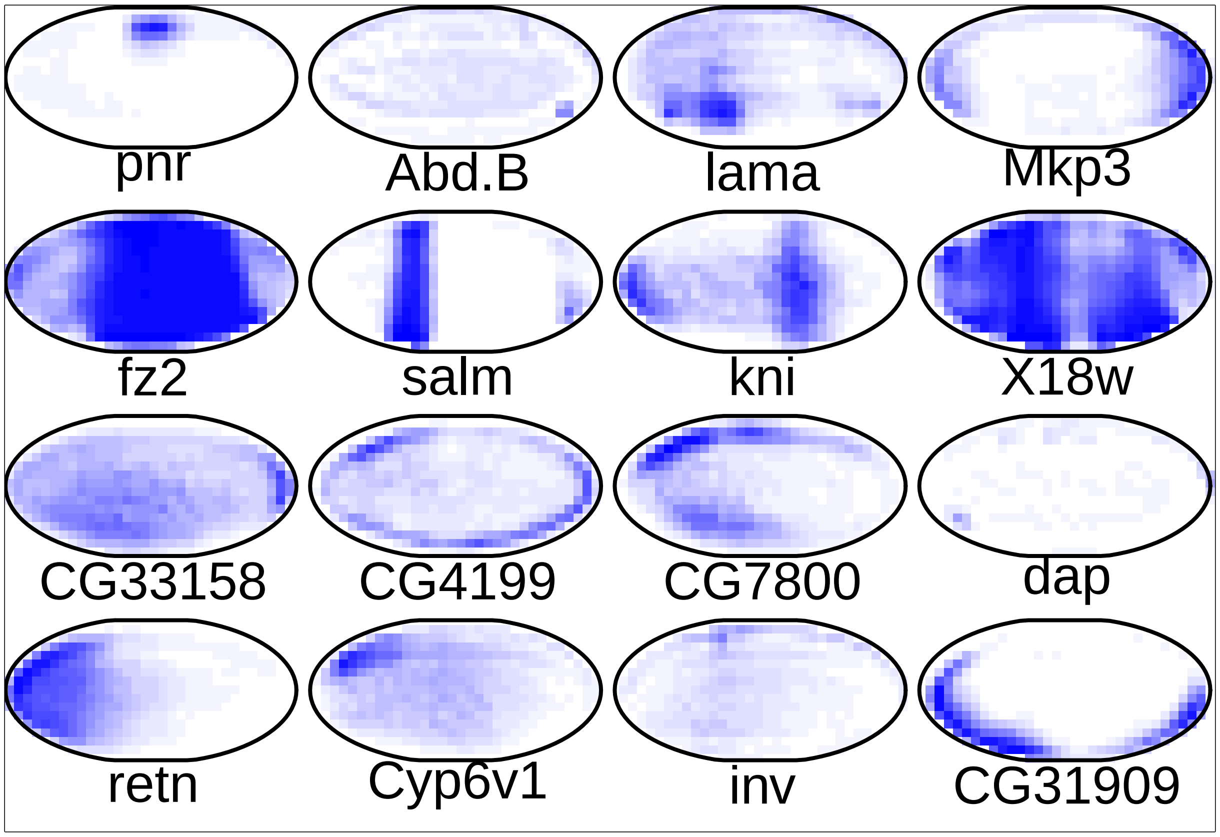
Problem 32: NMF of spatial gene expression patterns

Subproblem 1

```r
# load DrosophilaExpressions.rda
load("DrosophilaExpressions.rda")
```

Display the first 16 observations as elliptical images.

```r
imageBatchDisplay(V[, 1:16])
```



Factorization using `rank = 15`

Compute a factorization $V \approx \hat{V} = WH$ with `rank=15, seed=123` and default `method="brunet"`.

```
res <- nmf(V, rank = 15, method = "brunet", seed = 123)
```

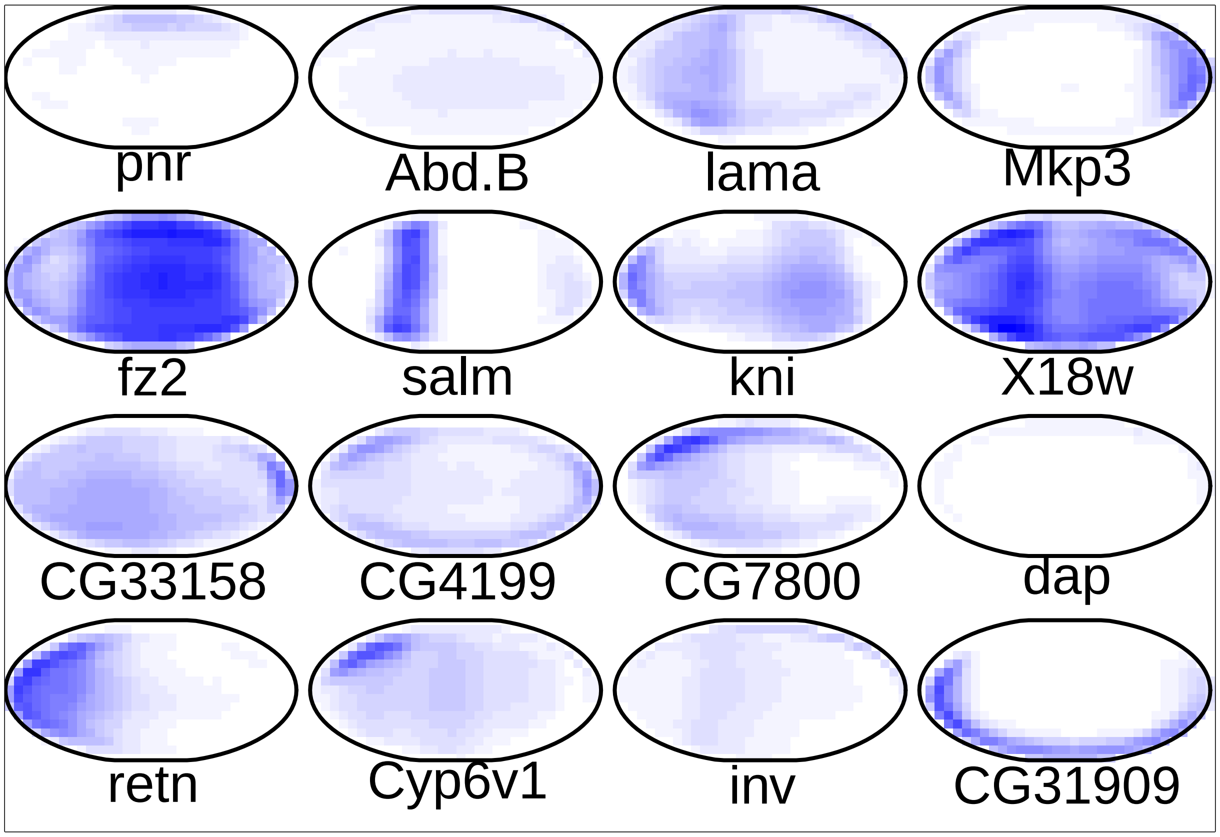
Report the KL divergence of the approximated data matrix $\hat{V}$.

```
summary(res)
```

```
##              rank  sparseness.basis  sparseness.coef  silhouette.coef
##        15.0000000         0.6164002        0.4756141        0.2960606
## silhouette.basis          residuals            niter              cpu
##         0.5188759      3626.1393612     2000.0000000       25.5390000
##           cpu.all              nrun
##        25.5390000         1.0000000
```

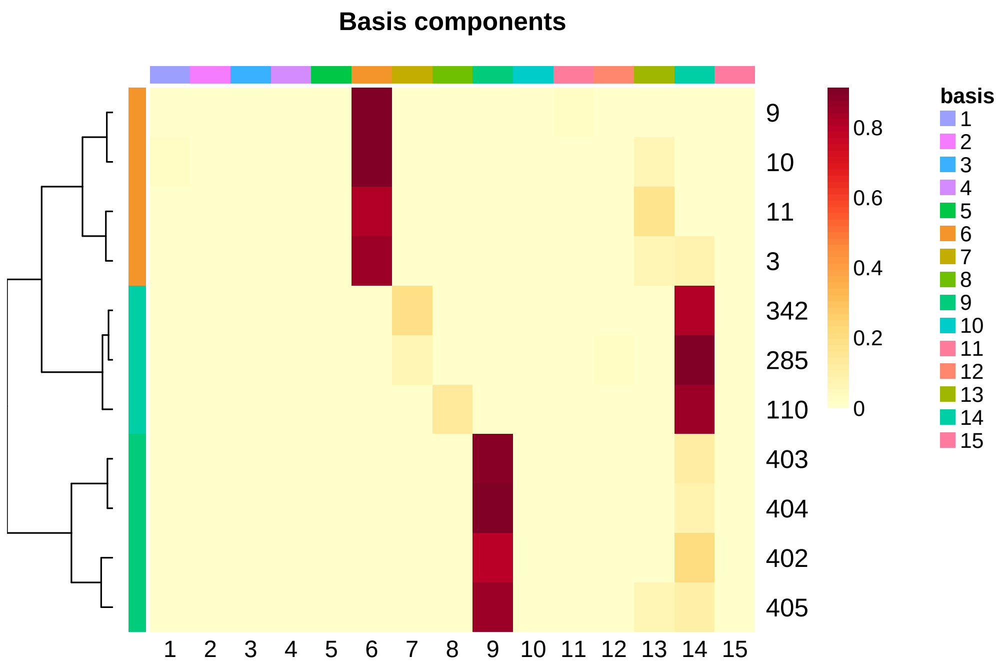
Display the first 16 columns of $\hat{V}$ as imagese.

```
V.hat <- fitted(res)
imageBatchDisplay(V.hat[, 1:16])
```



Display all computed basis patterns (columns of W).

```
# use basismap (from the documentation)
basismap(res, subsetRow=TRUE)
```

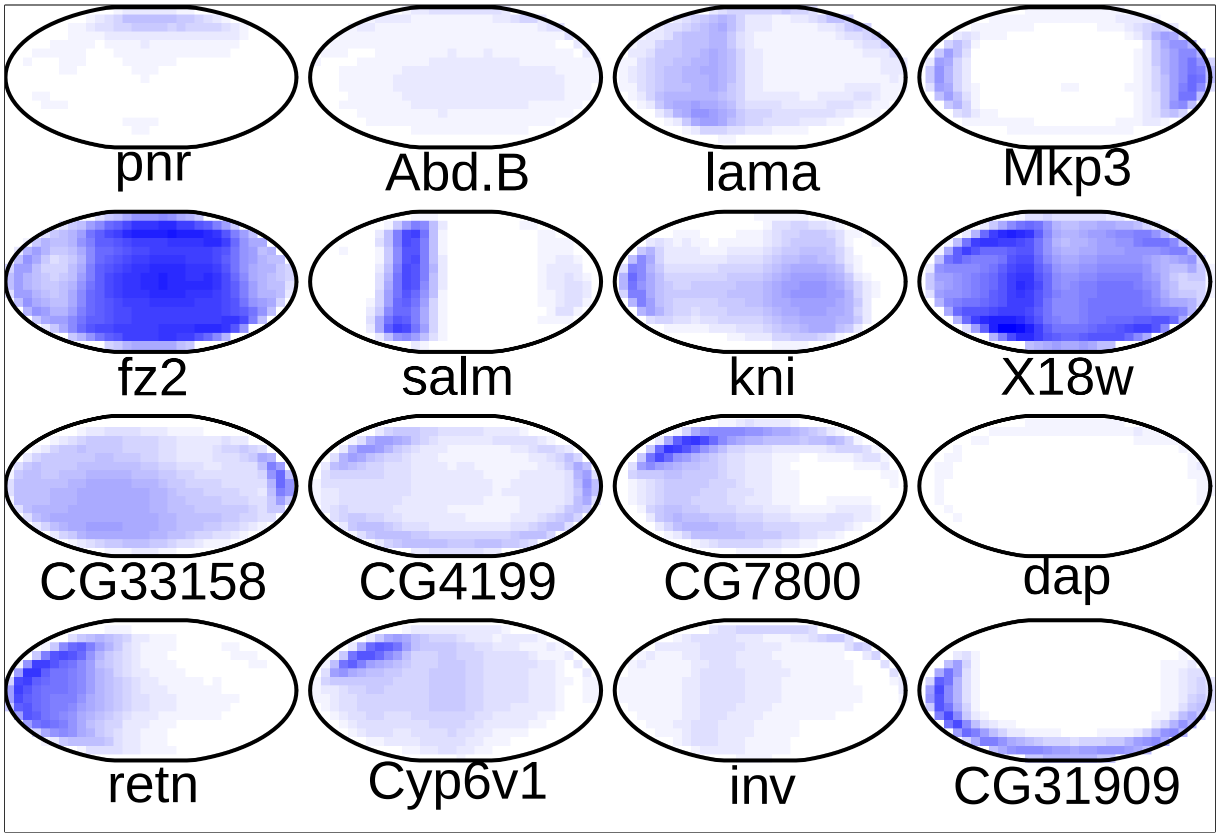


Factorization using rank = 10

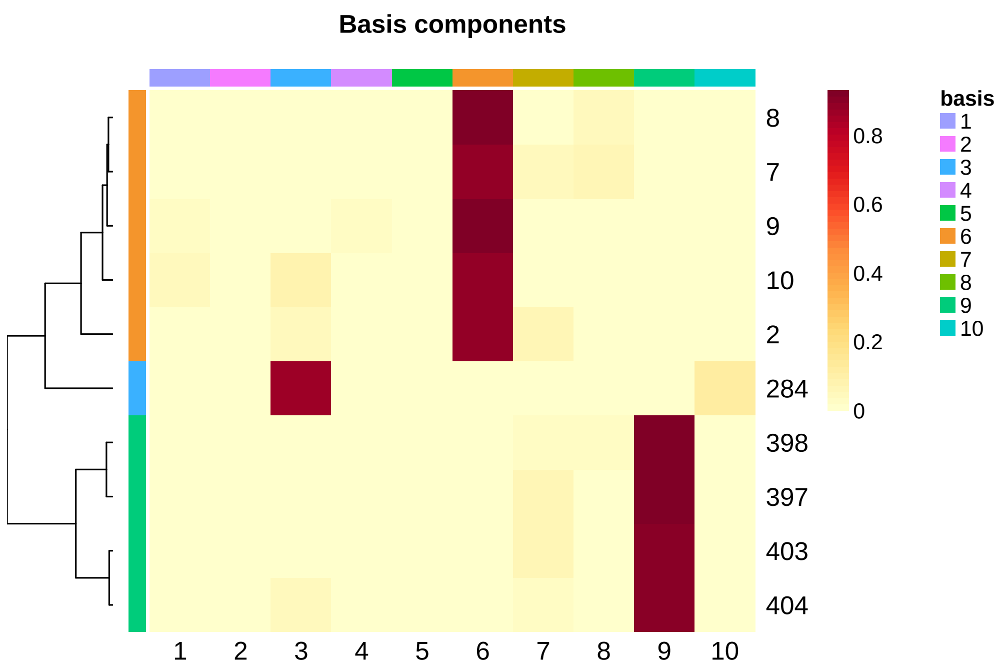
```r
res.10 <- nmf(V, rank = 10, method = "brunet", seed = 123)
# report KL divergence (note the "residuals")
summary(res.10)
```

```
##             rank sparseness.basis  sparseness.coef  silhouette.coef 
##       10.0000000        0.5346524        0.4762901        0.3740181 
## silhouette.basis        residuals            niter              cpu 
##        0.5392153     4901.2979563     2000.0000000       16.8190000 
##          cpu.all             nrun 
##       16.8190000        1.0000000
```

```r
# visualize V hat
V.hat.10 <- fitted(res)
imageBatchDisplay(V.hat.10[, 1:16])
```



```r
# display W
basismap(res.10, subsetRow=TRUE)
```

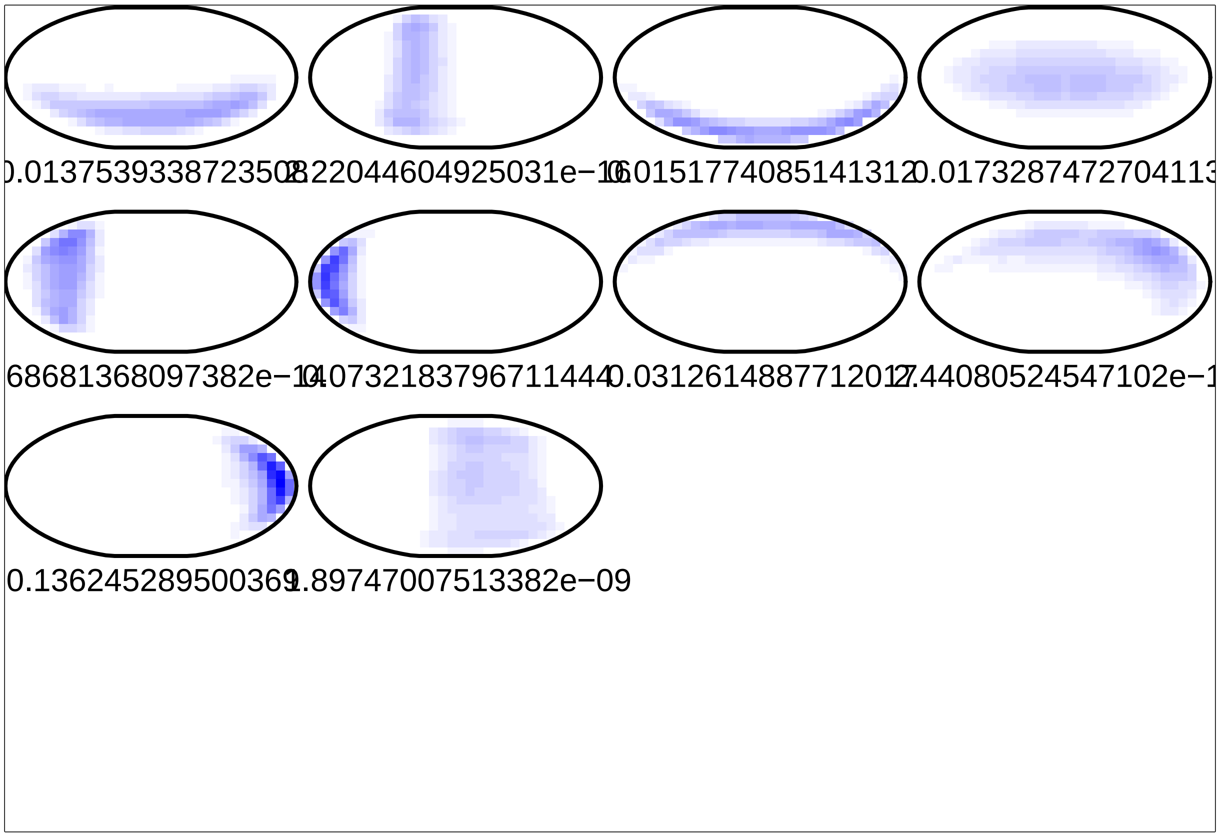


Subproblem 2

```r
W.10 <- basis(res.10)
H.10 <- coef(res.10)
```

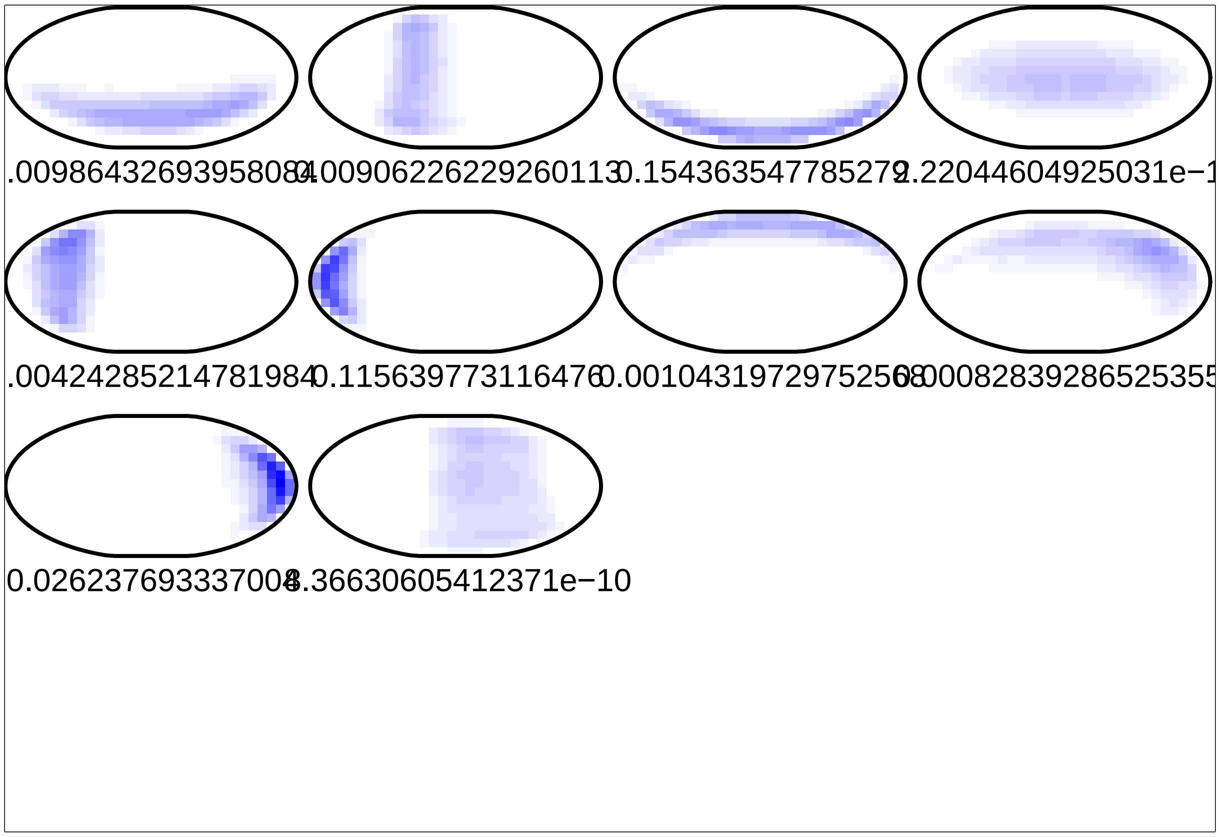
Gene `Mkp3`

```r
imageBatchDisplay(
  W.10,
  imgNames = H.10[, "Mkp3"],
  font = 1
)
```



Gene `CG31909`

```
imageBatchDisplay(
  W.10,
  imgNames = H.10[, "CG31909"],
  font = 1
)
```



Problem 33: Implementing NMF from scratch

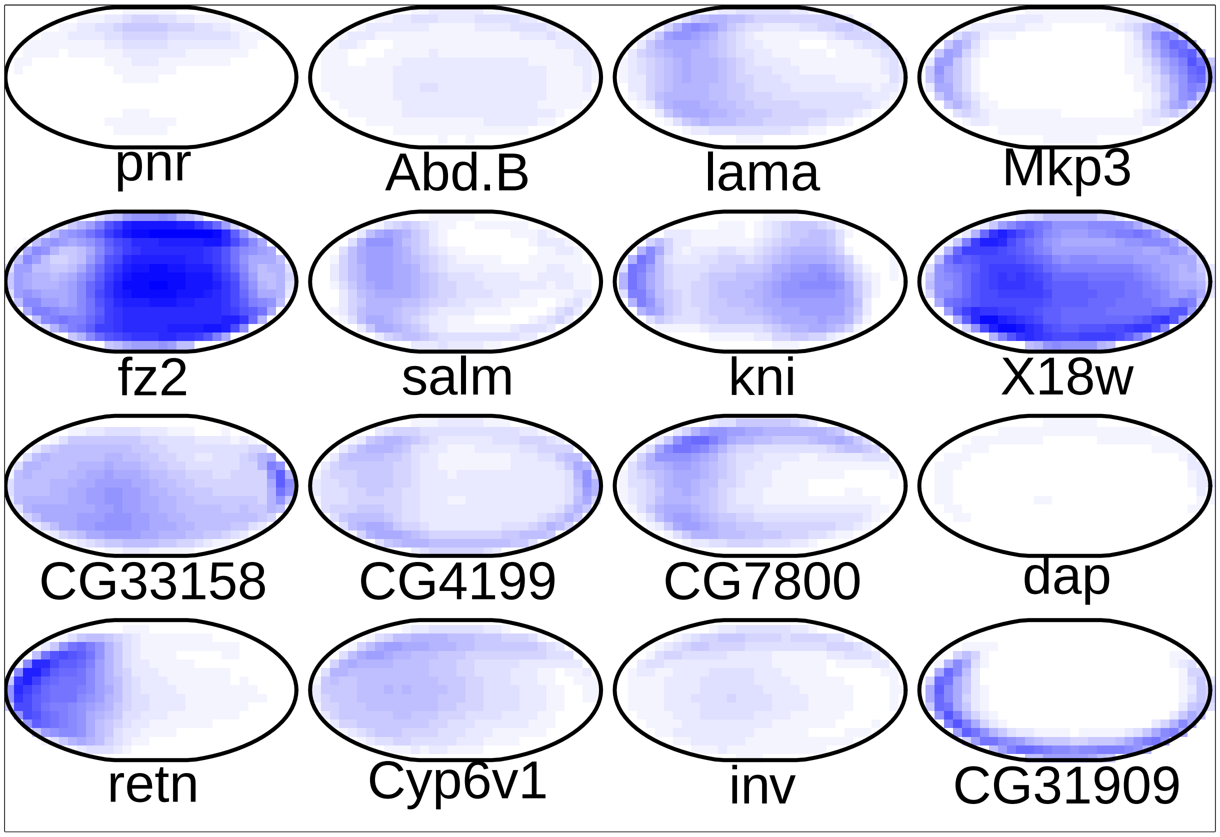
```
NMF <- function(V, k,
                W = NULL, H = NULL, max_iter = 10000,
                seed = 42) {
  set.seed(seed)
  M <- dim(V)[[1]]
  N <- dim(V)[[2]]
  if (is.null(W)) {
    W <- matrix(runif(M * k), nrow = M, ncol = k)
  }

  if (is.null(H)) {
    H <- matrix(runif(k * N), nrow = k, ncol = N)
  }

  W_new <- W
  H_new <- H
  to_log <- V /(W %*% H)
  to_log <- ifelse(to_log ≠ 0, log(to_log), 0)
  kl_div <- sum(V * to_log - V + W %*% H)
  prev_kl_div <- kl_div
  all_kl <- foreach (iter=1:max_iter, .combine = c) %do% {
    H_new <- H * (t(W) %*% (V/(W %*% H))) / (colSums(W))
    W_new <- W * ((V/(W %*% H_new)) %*% t(H_new)) / (rowSums(H_new))

    to_log <- V /(W_new %*% H_new)
    to_log <- ifelse(to_log ≠ 0, log(to_log), 0)
    kl_div <- sum(V * to_log - V + W_new %*% H_new)

    prev_kl_div <- kl_div
    W <- W_new
    H <- H_new
    kl_div
  }

  return(list(V.hat=W_new %*% H_new, W=W_new, H=H_new, KLdiv=kl_div, allKL=all_kl))
}
```

```
res <- NMF(V, 10, max_iter=1000)
```

```
res$KLdiv
```

```
## [1] 4861.621
```

```
imageBatchDisplay(res$V.hat[, 1:16])
```


pnr
Abd.B
lama
Mkp3
fz2
salm
kni
X18w
CG33158
CG4199
CG7800
dap
retn
Cyp6v1
inv
CG31909