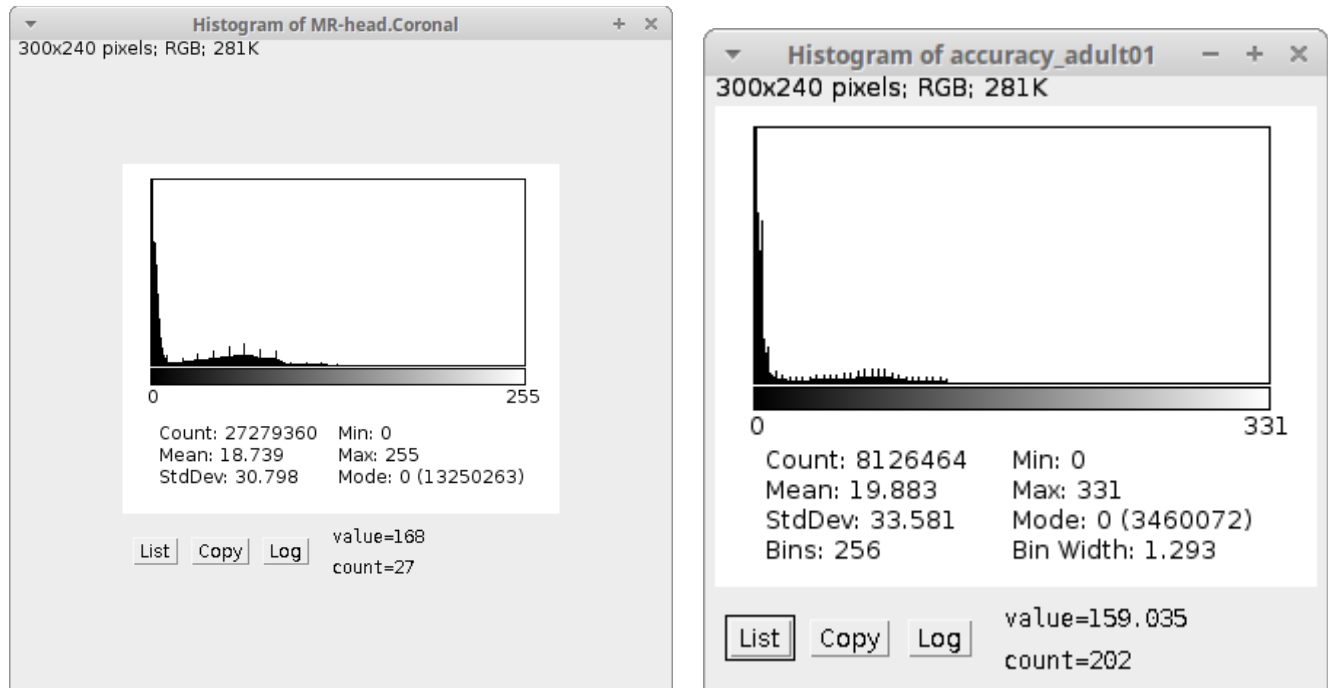


Compte rendu

Tristan Cossin

Question 2a.



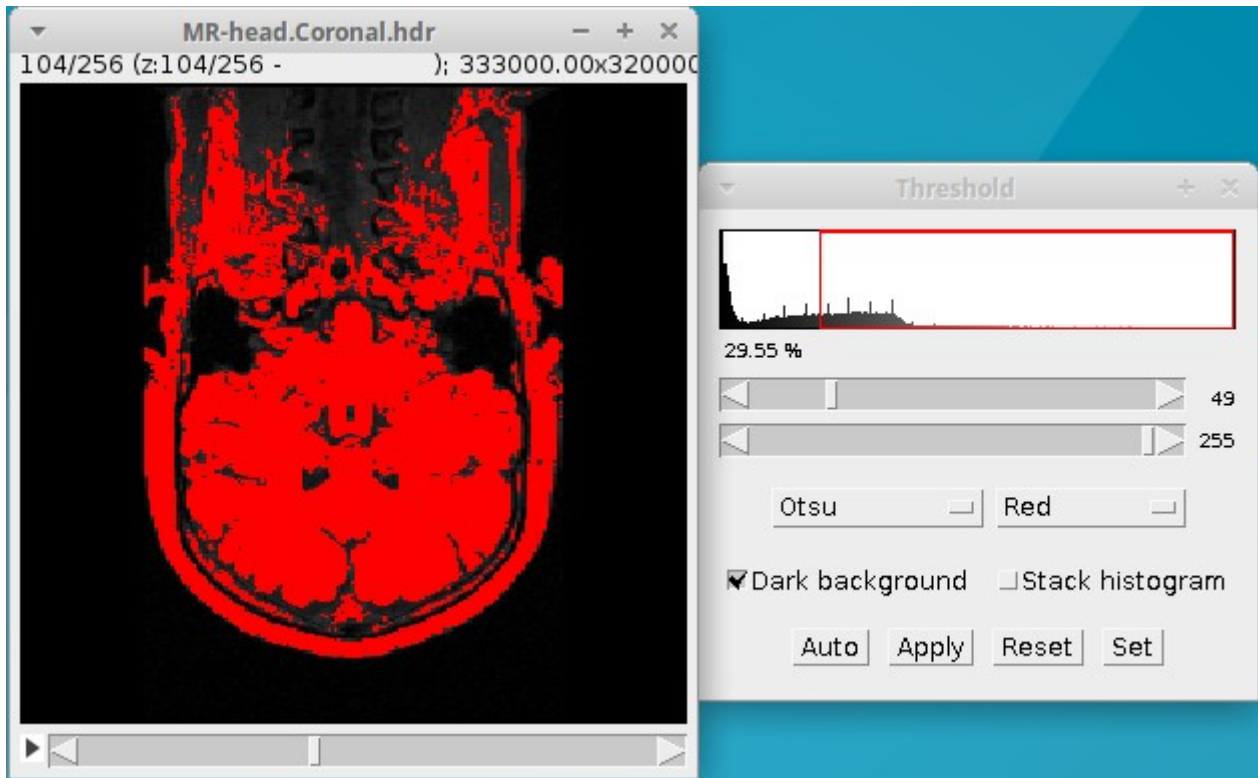
La forme de l'histogramme peut s'expliquer par le fait que les images sont principalement constituées de noir et que les différentes parties du crâne sont en niveau de gris d'où le fait que l'histogramme présente peu de blanc.

Question 2b.

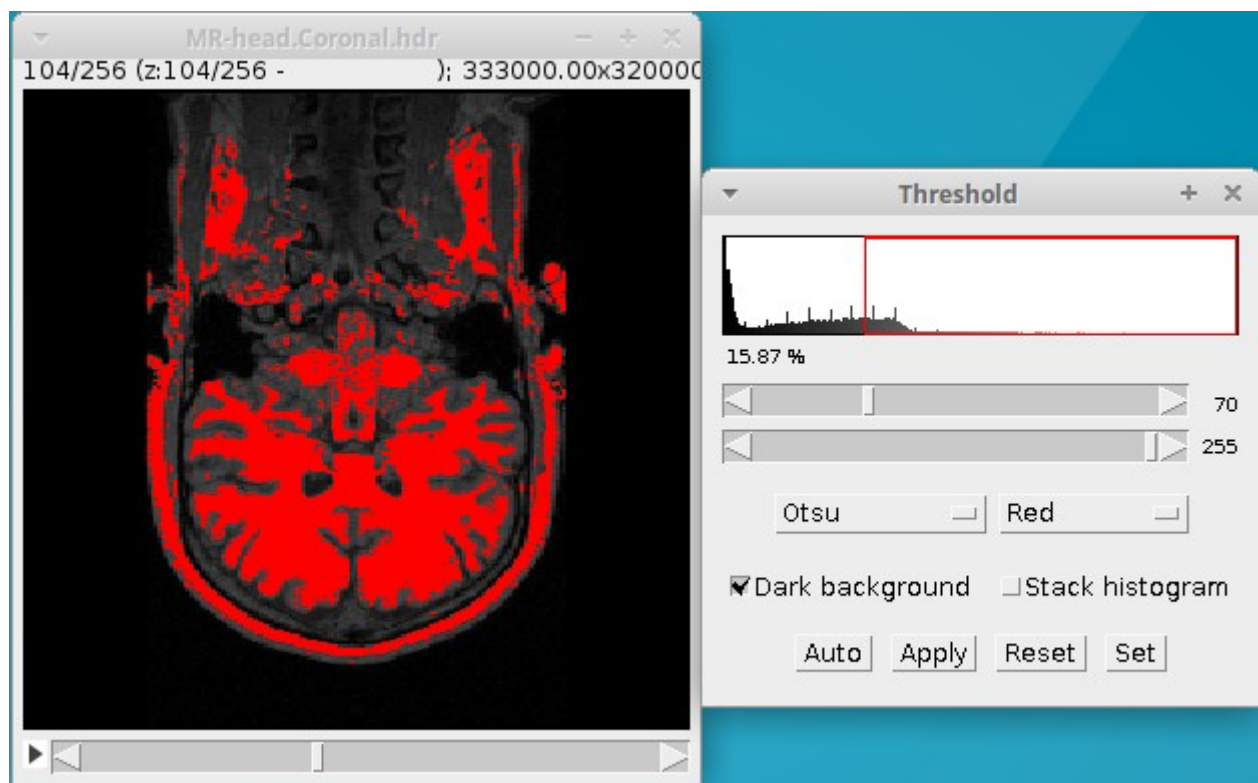
L'algorithme de clusterage de seuil d'Otsu recherche le seuil qui minimise la variance intra-classe, définie comme une somme pondérée des variances des deux classes.

En d'autres termes, le bouton Auto/Otsu permet de mettre en valeur les différents éléments constituant l'image.

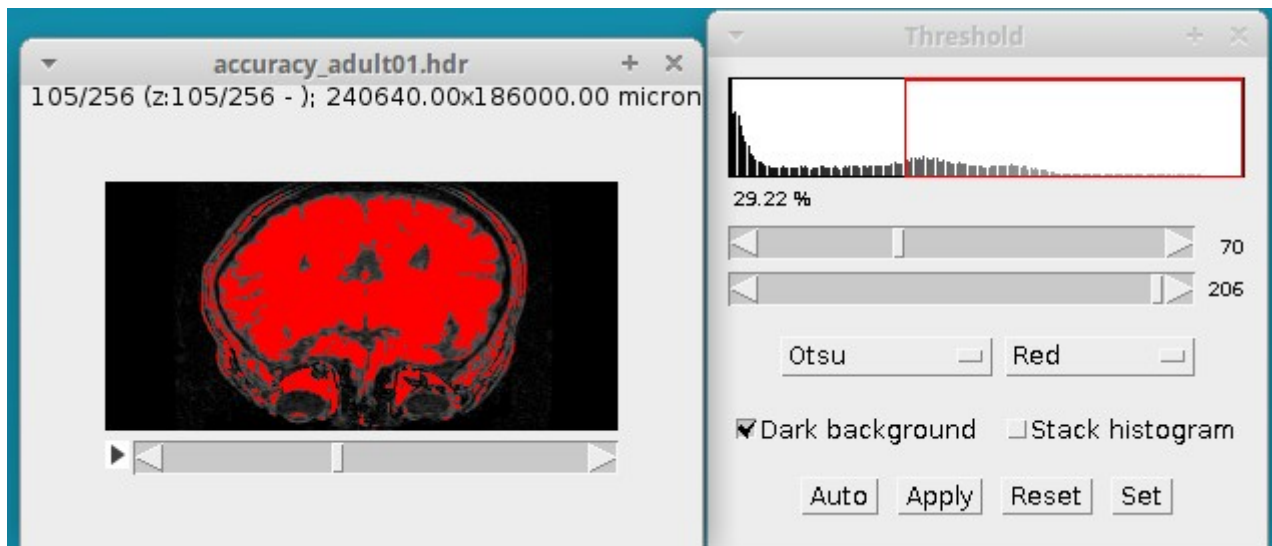
Question 3.



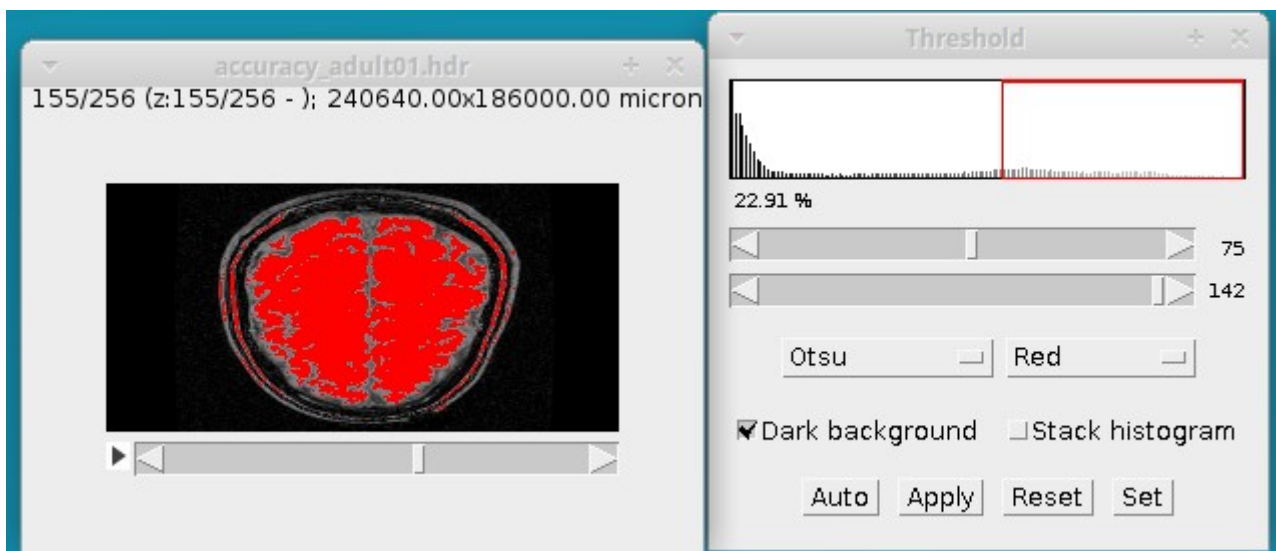
cerveau = 30 %



matière blanche = 16 %



cerveau = 29 %



matière blanche = 23 %

En seuillant à ces valeurs, on constate que l'on ne prend pas seulement le cerveau mais aussi d'autre partie du cerveau.

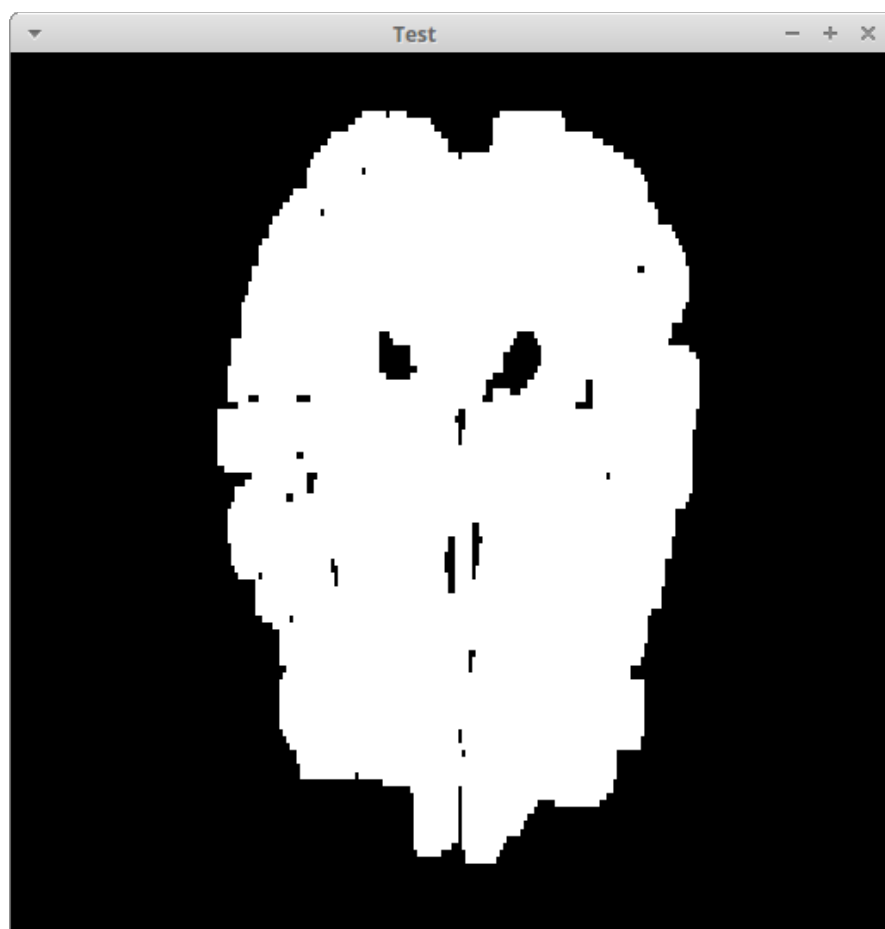
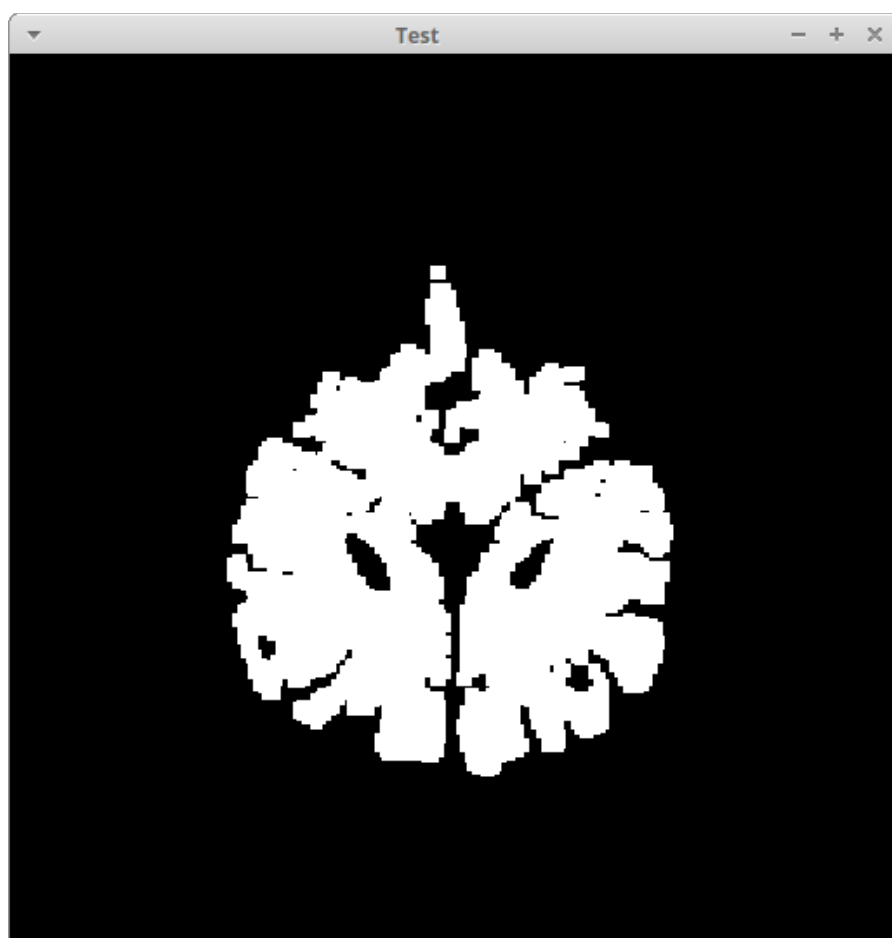
Question 4.

→ Pour commencer, on crée un masque binaire à partir d'un seuil, ce qui permet d'exclure le liquide cerebrospinal, l'air, etc...

→ A partir du masque obtenu, on applique une érosion afin de séparer le cerveau des autres éléments.

→ On sélectionne le plus gros élément.

→ Maintenant, on applique une dilatation pour compenser l'érosion du début.



```

img.threshold(50, false, true); //on seuil
img.erode(5, 5, 5); //on érode
img.label(true, 0); //on labélise

int histogramme[img.max()] = {0};

//on fais l'histogramme de l'image
for(int i = 0; i < img.width()-1; i++)
{
    for (int j = 0; j < img.height()-1; j++)
    {
        for (int k = 0; k < img.depth()-1; k++)
        {
            histogramme[*img.data(i, j, k)]++;
        }
    }
}

int seuil = 0;
int valeur = 0;

//on cherche la plus haute valeur de l'histogramme
// et on enregistre la valeur
for (int i = 1; i < img.max(); i++)
{
    if(histogramme[i] > seuil)
    {
        seuil = histogramme[i];
        valeur = i;
    }
}

//si le pixel n'a pas la valeur voulu par rapport
// à l'histogramme alors on met la valeur à 0
for(int i = 0; i < img.width()-1; i++)
{
    for (int j = 0; j < img.height()-1; j++)
    {
        for (int k = 0; k < img.depth()-1; k++)
        {
            if(*img.data(i, j, k) != valeur)
                img(i, j, k) = 0;
        }
    }
}

img.dilate(5, 5, 5); //on dilate

```