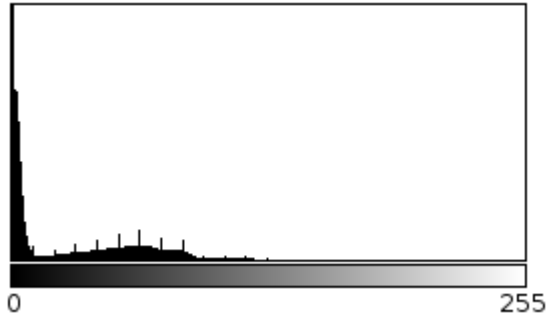


TP1 – Segmentation automatique du cerveau

I. Présentation du sujet

II. Analyses

2.



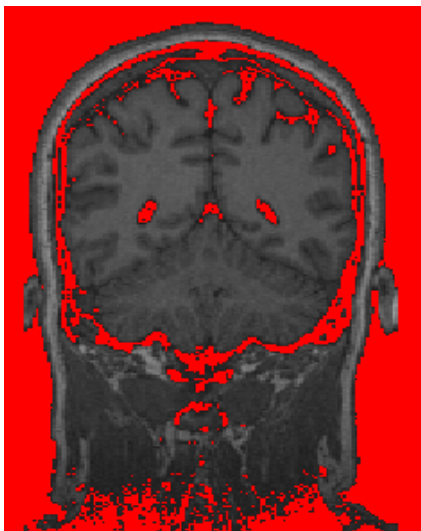
Histogramme de head.Coronal

La forme de cet histogramme est due à la grande majorité de noir dans l'image, étant donné que tous les éléments extérieurs à la tête ne sont pas pris en compte.

Les pics sont les zones redondantes, comme par exemple l'os de la boîte crânienne.

En appliquant l'algorithme d'Otsu donné par Fiji, on obtient une valeur de seuil égale à 75,63%. Cette valeur correspond à celle qui sépare le mieux les deux classes de couleur, en deux classes ayant une variance minimale.

3.



Dissociation du cerveau

Pour dissocier correctement le cerveau du reste de la boîte crânienne, on peut utiliser un seuil à 74,06%. En prenant une valeur inférieure, il restait trop de points de contact entre l'os et la matière visée, et les poches de liquide cérébral étaient incluses dans le cerveau.



Pour pouvoir dissocier la matière blanche du reste du cerveau, on utilise un seuillage à 91,23%. La matière grise restante au centre de l'image pourra être effacée en utilisant un algorithme d'érosion puis de dilatation.

Dissociation de la matière blanche

4. En entrée l'algorithme a besoin d'une image déjà seuillée. Ensuite, on va éroder l'image un nombre E de fois. Ensuite, on trie les zones restantes par ordre décroissant, en fonction de celles qui contiennent le plus d'éléments. Une fois ce tri fait, on va dilater les zones les plus importantes.

```
int main(int argc, char *argv[]){
    int cpt = 100;
    int thresholdValue = 70;
    int erosionCpt = 5;

    CImg<> img("MR_head.Coronal.hdr"), res = img.get_threshold(thresholdValue);
    CImgDisplay display(img, "MR_head.Coronal.hdr");

    bool algo = false;

    while (!display.is_closed()) {
        if(display.is_keyESC())
            break;
        if(display.is_keyA())
            algo = true;
        if(display.wheel()){
            cpt += display.wheel();
            display.set_wheel();
        }

        if(display.is_keyPADADD()){
            thresholdValue = (thresholdValue + 1) % 255;
            res = img.get_threshold(thresholdValue);
        }
        if(display.is_keyPADSUB()){
            thresholdValue = (thresholdValue - 1) % 255;
            res = img.get_threshold(thresholdValue);
        }
        display.display(res.get_slice(cpt));
    }
}
```

```

if(algo){
    std::cout << "Start erosion" << std::endl;
    for(int i = 0; i < erosionCpt; ++i)
        res.erode(2, 2, 2);

    res.label(0, 0);
    int histo[256];
    memset(histo, 0, sizeof(histo));

    for(int i = 0; i < img.width(); ++i){
        for(int j = 0; j < img.height(); ++j){
            for(int k = 0; k < img.depth(); ++k){
                histo[(int)res(i, j, k)]++;
            }
        }
    }
    int max = 0;
    int maxI = 0;
    for(int i = 0; i < 256; ++i){
        if(histo[i] > max){
            max = histo[i];
            maxI = i;
        }
    }
    for(int i = 0; i < img.width(); ++i){
        for(int j = 0; j < img.height(); ++j){
            for(int k = 0; k < img.depth(); ++k){
                if(res(i, j, k) == maxI)
                    res(i, j, k) = 0;
                else
                    res(i, j, k) = 255;
            }
        }
    }
    for(int i = 0; i < erosionCpt; ++i)
        res.dilate(2, 2, 2);

    display.display(res.get_slice(cpt));
    algo = false;
}
display.wait();
}
return 0;
}

```

En exécutant l'algorithme, on obtient l'image ci-dessous, qui représente le cerveau, après érosion, sélection des zones optimales, puis dilatation.



MR_head.Coronal