

# Les Réparations de Bêta-Amyloïde et L'Hypothèse Amyloïde

-Jojo

## Introduction

La démence affecte approximativement 50 millions personnes dans le monde et la syndrome est caractérisée par: "une dégradation de la mémoire, du raisonnement, du comportement et de l'aptitude à réaliser les activités quotidiennes". Ces effets peuvent poser un problème pour quelqu'un qui a la démence et aussi peut affecter leur relations avec sa famille et ses amis. Donc, les effets de la démence n'est pas limité aux personnes qui l'entourent.

L'Alzheimer est un type de démence, et constitué 60–70% des cases, donc c'est le forme primaire. Il y a d'autres formes et causes de démence; notamment, la démence vasculaire et la démence à corps de Lewy. Mais, le tracé entre celles-ci n'est pas tellement ou toujours clair, donc c'est plus inclusif de dire que un case est un mélange de ces formes primaires. Dans un sens scientifique, la démence est caractérisée par la dégradation des parts de cerveau.

Nous focalisons sur L'Alzheimer parce qu'il y a un bon quantité de données sur les protéines au centre du syndrome. Spécifiquement, Bêta-amyloïde, une chaîne polypeptidique, qui forment après le traitement de Amyloïde Bêta-protéine précurseur (APP). Mas trop est connu sur APP, mais, c'est un fil de code de DNA. Donc, APP est justement quelques acides aminés, et il y a juste 21 de cela. Un peut-être plus concrète un séquence des «A», «C», «G», ou «T». Je l'ai retrouvé pour APP; il y a 922 «A», 745 «C», 868 «G», et 819 «T». Donc, il y a 3354 bases qui forment  $3354/3 = 1118$  acides aminés (tu peux le regarde dans le appendix).

En brevet, quand cette séquence est traitée un protéine et forme, mais aussi résultat d'une résidu... de Bêta-amyloïde. Il y a plus à dire sur le motif de structure dans protéines et questions sur comment et pourquoi on arrive à une réparation des residues particulière; ... mais peut-être un autre fois. Dans le monde Alzheimer's il y a un "Hypothèse Amyloïde":

$H_0$ : L'Accumulation de Bêta-amyloïde est la cause primaire de Alzheimers.

Équipé avec des données et des outils statistiques, on peut tester l'hypothèse.

## Le Donnée

Il y a de multiples sources de données sur Les Alzheimers, mais beaucoup sont petits or n'a pas beaucoup d'information (ils n'est pas très riche). Donc, j'ai appliqué pour accès les données de la Alzheimer's Disease Neuroimaging Initiative (ADNI), et faciliter

technologiquement par le laboratoire de la neuroimagerie à l'Université de Californie de sud. Ils m'acceptent le 20 Mai, donc toute mon analyse est préliminaire.

Les objectifs de le ADNI sont grand, mais réalisables:<sup>1</sup>

- Détecter L'Alzheimer plus tôt que possible (avant-démence) et identifier la progression avec des biomarqueurs.
- Supporté avancements en méthodes nouvelles pour intervenir, prévenir, traiter, et diagnostiquer L'Alzheimer.
- Administrer un programme innovant pour accès données, pour tous des scientists dans le monde.

Beaucoup des centres de recherche essayent de traiter ou prévenir L'Alzheimer, donc le deuxième est un peu attendu. La première est un peu plus fraîche. Même si c'est difficile de traiter ou prévenir L'Alzheimer, on peut le traquer et comprendre les modèles de progression. Et si on peut *compris* assez pour projeter l'avenir, on peut expérimenter avec des approches au traitement et prévention pour évaluer leur efficacité (ça nécessite des méthodes statistiques principales qui s'appellent... conception expérimentale (en anglais, c'est "experimental design")).

Il y a plus que 100 Observables, mais pour une analyse simple, on utilise celui-là.

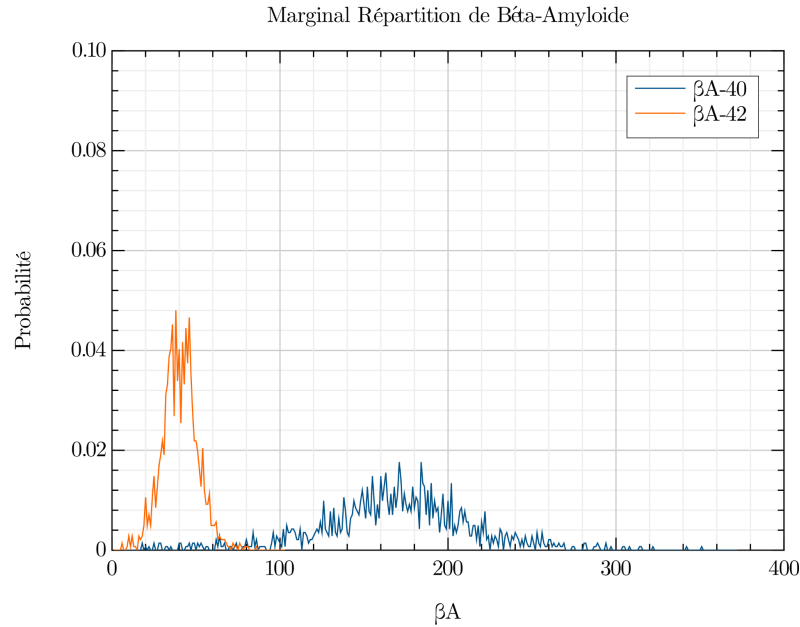
Observable	Description
RID	Nombre qui identifie un patient (ça c'est anonyme)
VISCODE	Code de visit («bl»=point de départ, «m12»= bl+12 mois, «m24», «m36», «m48»)
Aβ40	Béta-Amyloïd 1-40 dans plasma
Aβ42	Béta-Amyloïd 1-42 dans plasma
DXCURRENT	Diagnostic (NL=normal, MCI=déficiência cognitive modere, AD=Alzheimers)

Premièrement, j'ai calculé quelques statistiques pour chaque observable. Ils sont justement des nombres (pas trop informatifs), donc j'ai visualisé les réparations de chaque observable.

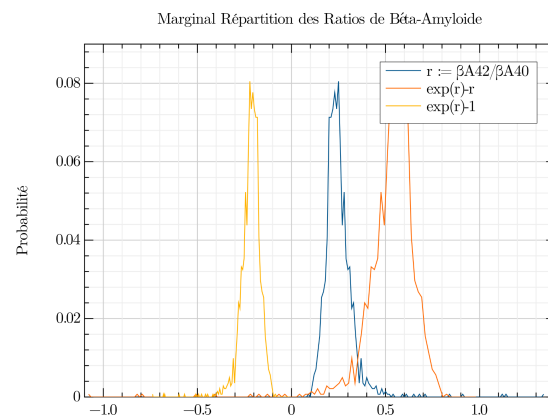
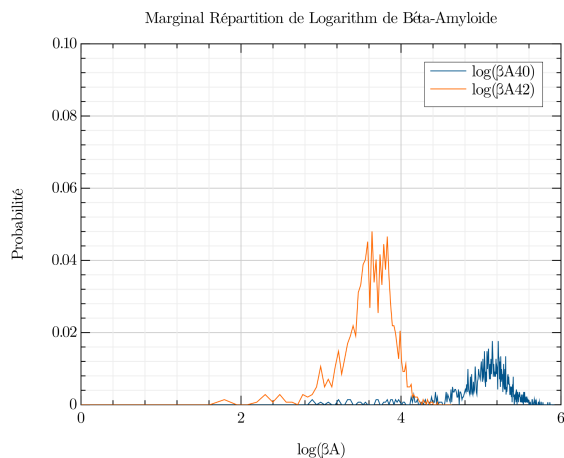
Statistic	Aβ40	Aβ42	Aβ42/Aβ40
Min.	17.8	5.9	0.0869
Max.	351.1	85.7	1.3410
Moyenne	169.1649	40.4773	0.2511
Écart type	46.0782	10.5938	0.0826

---

<sup>1</sup> Un traduction libéral de <http://adni.loni.usc.edu/about/>



Voici les réparations de Bêta-Amyloïde 40 et Bêta-Amyloïde 42. Bêta-Amyloïde 40 est plus typique, mais avec ça, viens plus d'écart. Au l'autre main, Bêta-Amyloïde 40 est moins prévalent, et a moins d'écart. Le deux réparations est tellement distinguée, mais le deux a des similar ratios Moyenne-Écart type. Pour Bêta-Amyloïde 40 et 42,  $\mu/\sigma$  sont 3.6712 et 3.8208. Ça implique que c'est *possible* que des distributions viennent du même processus hazard. Aussi, le ratio écart-moyenne pour Bêta-Amyloïde 40 est  $\sigma_{40}^2/\mu_{40}=12.5510$ , et au carré, c'est approximativement écart-moyenne de Bêta-Amyloïde 42,  $\sigma_{42}^2/\mu_{42}=154.6576$  ( $(\sigma_{40}^2/\mu_{40})^2=12.5510^2=157.527601$ ). Ça implique que le processus hazard que ils (peut-être) partagent est proportionnellement l'inverse du nombre residue. L'idée est que, nous pouvons identifier un "micro"-réparation de Bêta-Amyloïde pour dériver le réparations prévus de  $A_k$  (pour tout k possible). C'est intéressant; quand k est moins de 15 les biochimiste disent que le protein est un "oligopeptide", quand  $15 < k < 50$  ils disent que c'est un "polypeptide", et quand c'est plus grand que 50 c'est justement un protéine.

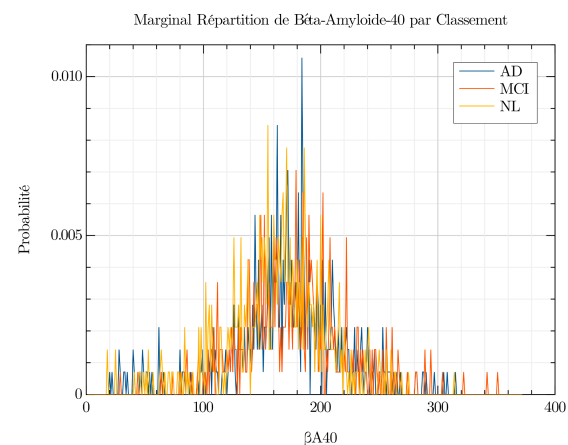
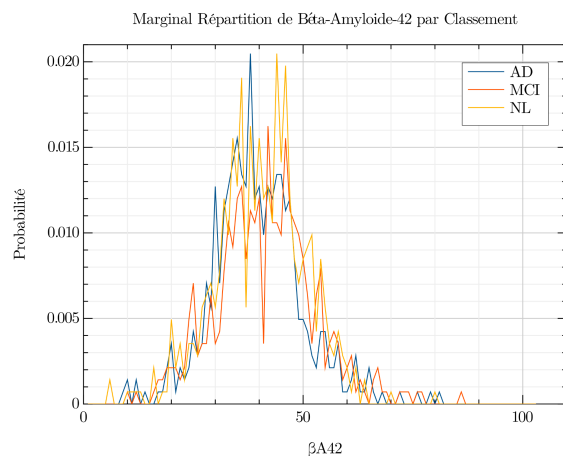


C'est difficile d'identifier quelques "micro"-réparations parce que, en général, quand  $k$  est un nombre égal, les réparations incluent les symétries des asymétries. Ça nécessite des techniques dans Analyse de Fourier (pour un autre jour).

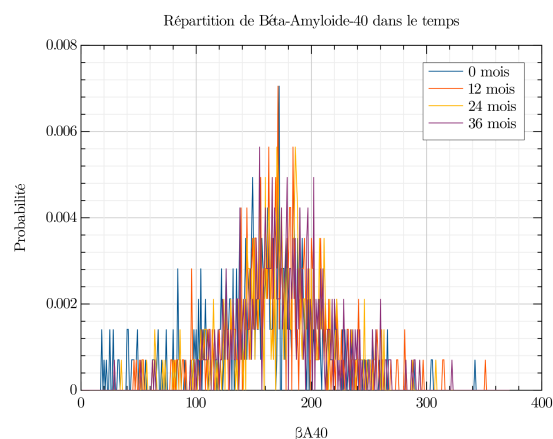
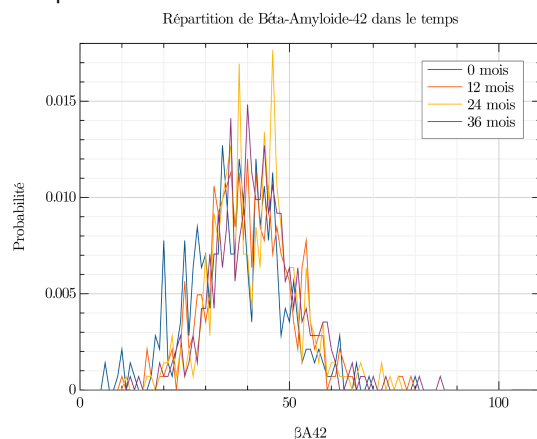
Pour plus de résultats concrets, on retourne à l'Hypothèse d'Amyloïde. Rappel,

$H_0$ : L'Accumulation de Bêta-amyloïde est la cause primaire de Alzheimers.

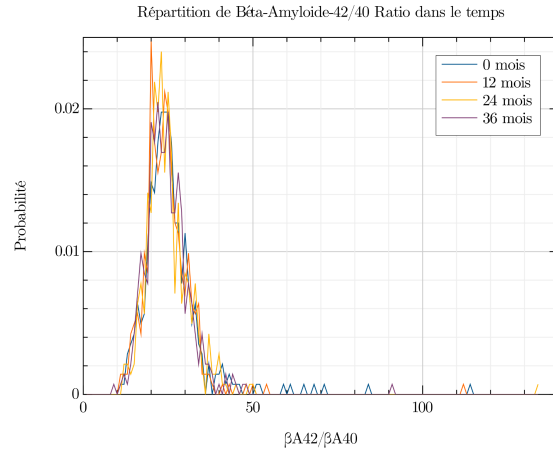
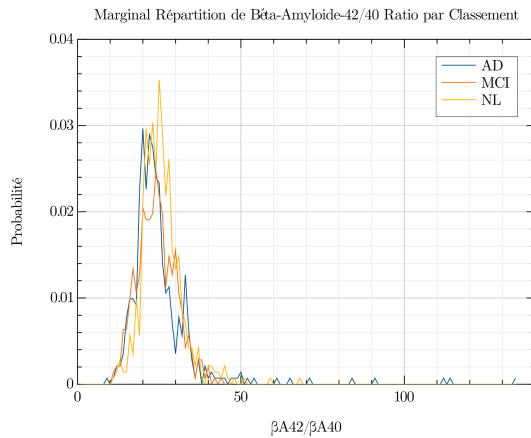
Le première chose qu'on peut regarder pour analyse  $H_0$  sont des réparations conditionnel sur des diagnostics (NL=normal, MCI=déficiance cognitive modérée, AD=Alzheimers). Celles-ci sont tracées en bas.



C'est difficile de tant discerner. Voir qu'il y a un haut forme des réparations, et aussi des petites variations pour chaque réparation. Est-ce que c'est un caractisme des réparations conditionnelles de diagnostic? Probablement, non. Voir les réparations conditionnelle sur temps. Ils ont les mêmes tendances.



Ces tendances n'est pas très bizarre parce que le donné est discret mais aussi bruyant (et donc aussi sont des réparations). Les réparations de la ratio,  $A\beta_{42}/A\beta_{40}$ , est plus belle. En même temps, les réparations pour chaque classe, et dans le temps, sont très similaires.



Donc, j'ai créé un essai pour évaluer les différences entre les réparations conditionnelle. Laisse,  $c_{NL}$  être la probabilité qu'on voit un personnage "Normal", et la même définition pour  $c_{MCI}$  et  $c_{AD}$ . Aussi, laisse  $P_{NL}$  être le réparation sur les observables qui reste (temps,  $A\beta 40$ ,  $A\beta 42$ ) conditionnelle qu'on voit un personnage, et la même définition pour  $P_{MCI}$  et  $P_{AD}$ . Ces réparations ont trois dimensions, donc vous pouvez les penser comme un prisme rectangulaire. Le réparation empirique moins le diagnostic observable est:  $U = P_{NL} + P_{MCI} + P_{AD}$ . On pris, pour notre statistique essai,  $k := \frac{1}{2}(c_{NL} |P_{NL}-U| + c_{MCI} |P_{MCI}-U| + c_{AD} |P_{AD}-U|)$ , et le nomme "variation-totale conditionnel".  $k$  est entre 0 et 1, et le probabilité que les réparations  $P_{NL}, P_{MCI}, P_{AD}$ , viennent de la même réparation est plus grande que  $1/(1+k)$ .

Le probabilité que les réparations, de temps,  $A\beta 40$ ,  $A\beta 42$ , conditionnelle sur classe soient le même est plus grande que 0.9999999999999976 (le statistic  $k$  est  $2.50 \times 10^{-15}$ ). Et j'ai trouvé le même résultat si on a les réparations de temps,  $A\beta 42/A\beta 40$ .

Donc pour tout degré de confiance,  $1-\alpha$ , moins grande 0.9999999999999976 nous pouvons rejeter l'hypothèse  $H_0$  et trouvons en faveur de  $H_1 = \text{« pas } H_0 \text{ »}$ .

## Citations:

Alzheimer's Disease Neuroimaging Initiative, <http://adni.loni.usc.edu/>.

Fandos, Noelia et al. "Plasma amyloid  $\beta$  42/40 ratios as biomarkers for amyloid  $\beta$  cerebral deposition in cognitively normal individuals." *Alzheimer's & dementia* (Amsterdam, Netherlands) vol. 8 179-187. 12 Sep. 2017, doi:10.1016/j.dadm.2017.07.004

Organisation Mondiale de la Sante, La démence, <https://www.who.int/fr/news-room/fact-sheets/detail/dementia>.

Qiu, T, Liu, Q, Chen, Y-X, Zhao, Y-F, and Li, Y-M (2015), A $\beta$ 42 and A $\beta$ 40: similarities and differences. *J. Pept. Sci.*, 21, 522– 529. doi: 10.1002/psc.2789.

Westmark, C. What's hAPPening at synapses? The role of amyloid  $\beta$ -protein precursor and  $\beta$ -amyloid in neurological disorders. *Mol Psychiatry* 18, 425–434 (2013). <https://doi.org/10.1038/mp.2012.122>

DNA Sequence pour AAP (<https://www.ebi.ac.uk/ena/browser/api/fasta/Y00264.1>) :

```
AGTTTCCTCGGCAGCGGTAGGCGAGAGCACGCGGAGGAGCGTGC GCGGGGGCCCCGGGAGACGGCGCGGTGGCGGCGCGGGCAGAGCAAG
GACGCGGCGGATCCCACTCGCACAGCAGCGCACTCGGTGCCCGCGCAGGGTCGCGATGCTGCCCGGTTTGGCACTGCTCCTGCTGGCCGCTG
GACGGCTCGGGCGCTGGAGGTACCCACTGATGGTAATGCTGGCCTGCTGGCTGAACCCAGATTGCCATGTTCTGTGGCAGACTGAACATGCACAT
GAATGTCCAGAATGGGAAGTGGGATTGAGATCCATCAGGGACCAAAACCTGCATTGATACCAAGGAAGGCATCCTGCAGTATTGCCAAGAAGTCTAC
CCTGAAGTGCAGATACCAATGTGGTAGAAGCCAACCAACAGTGACCATCCAGAAGTGGTGCAAGCGGGGCGGCAAGCAGTGCAAGACCCATCC
CCACTTTGTGATTCCCTACCGCTGCTTAGTTGGTGAGTTTGAAGTGATGCCCTTCTCGTTCTGACAAGTGCAAATCTTACACCAGGAGAGGATGGA
TGTTTGCGAAACTCATCTTCACTGGCACACCGTCGCCAAAGAGACATGCAGTGAGAAGAGTACCAACTTGCATGACTACGGCATGTTGCTGCCCTGC
GGAATTGACAAGTTCCGAGGGGTAGAGTTTGTGTGTTGCCACTGGCTGAAGAAAGTGACAATGTGGATTCTGCTGATGCGGAGGAGGATGACTCG
GATGCTGGTGGGGCGGAGCAGACACAGACTATGCAGATGGGAGTGAAGACAAAGTAGTAGAAGTAGCAGAGGAGGAAGAAGTGGCTGAGGTGG
AAGAAGAAGAAGCCGATGATGACGAGGACGATGAGGATGGTGATGAGGTAGAGGAAGAGGCTGAGGAACCCCTACGAAGAAGCCACAGAGAGAAC
CACCAGCATTGCCACCACCACCACCACACAGAGTCTGTGGAAGAGGTGGTTCGAGTTCCTACAACAGCAGCCAGTACCCCTGATGCCGTTGA
CAAGTATCTCGAGACACTGGGGATGAGAATGAACATGCCATTTCCAGAAAGCCAAGAGAGGCTTGAAGCCAAAGCACCAGAGAGAAATGTCCCA
GGTCATGAGAGAATGGGAAGAGGCAGAACGTCAGCAAAAGAACTGCCTAAAGCTGATAAGAAGGCAGTTATCCAGCATTTCCAGGAGAAAGTGGA
ATCTTTGGAACAGGAAGCAGCAACGAGAGACAGCAGCTGGTGGAGACACATGGCCAGAGTGGAAGCCATGCTCAATGACCCGCCGCCCTGG
CCCTGGAGAACTACATCACCCTCTGCAGGCTGTTCTCTCGGCCCTGTCACGTGTTCAATATGCTAAAGAAGTATGTCCGCGCAGAACAGAAAGGA
CAGACAGCACACCCTAAAGCATTTGAGCATGTGCGCATGGTGGATCCCAAGAAAGCCGCTCAGATCCGGTCCAGGTTATGACACACCTCCGTGTG
ATTTATGAGCGCATGAATCAGTCTCTCTCCCTGCTCTACAACGTGCCTGCAGTGCCGAGGAGATTGAGGATGAAGTTGATGAGCTGCTTCAGAAAGA
GCAAACTATTGAGATGACGCTTGGCCAACATGATTAGTGAACCAAGGATCAGTTACGGAAACGATGCTCTCATGCCATCTTTGACCGAAACGAAAA
CCACCGTGGAGCTCCTTCCCGTGAATGGAGAGTTGAGCCTGGACGATCTCCAGCCGTGGCATTCTTTTGGGGTGAAGTCTGTGCCAGCCAACACAG
AAAACGAAGTTGAGCCTGTTGATGCCCCCCTGCTGCCGACCGAGGACTGACCACTCGACCAAGGTTCTGGGTTGACAAATATCAAGACGGGAGAGA
TCTCTGAAGTGAAGATGGATGCAGAATTCGACATGACTCAGGATATGAAGTTCATCATCAAAAATTGGTGTCTTTGCAGAAGATGTGGGTTCAAACA
AAGGTGCAATCATTGGACTCATGGTGGGCGGTGTTGTATGACGACAGTGATCGTATCACCTTGGTGATGCTGAAGAAGAAGACAGTACACATCCATT
CATCATGGTGTGGTGGAGGTTGACGCCGCTGTACCCAGAGGAGCGCCACCTGTCCAAGATGCAGCAGAACGGCTACGAAAATCCAACCTACAAG
TTCTTTGAGCAGATGCAGAACTAGACCCCGCCACAGCAGCCTCTGAAGTTGGACAGCAAAACCATTTGCTTCACTACCCATCGGTGTCCATTTATAGA
ATAATGTGGGAAGAAACAAACCCGTTTATGATTACTCATTATCGCCTTTTGACAGCTGTGCTGTAACACAAGTAGATGCCTGAACCTGAATTAATCCAC
ACATCAGTAATGTATTCTATCTCTTTACATTTTGGTCTCTATACTACATTATTAATGGGTTTGTGTACTGTAAAGAATTAGCTGTATCAAACTAGTGCAT
GAATAGATTCTCTCCTGATTATTTATCACATAGCCCTTAGCCAGTTGATATTAATCTTTGTGGTTTGTGACCCAATTAAGTCTACTTTACATATGCTTTAAG
AATCGATGGGGGATGCTTCATGTGAACGTGGGAGTTGACGTGCTTCTTGGCTAAGTATTCTTTCTGATCACTATGCATTTTAAAGTTAAACATTTTT
AAGTATTTGAGATGCTTTAGAGAGATTTTTTCCATGACTGCATTTTACTGTACAGATTGCTGCTTCTGCTATATTTGTGATATAGGAATTAAGAGGATACA
CAGTTTTGTTTCTCGTGCTGTTTTATGTGCACACATTAGGCATTGAGACTTCAAGCTTTCTTTTTTGTCCACGTATCTTTGGGTCTTTGATAAAGAAA
AGAATCCCTGTTTATTGAAGCACTTTACGGGGCGGGTGGGGAGGGGTGCTCTGCTGGTCTTCAATTACCAAGAATTCTCCAAAACAATTTCTGCAG
GATGATTGTACAGAATCATTGCTTATGACATGATCGCTTTTACTGTATTACATAAATAAATAAATAAATAAACCCTGGGCAAGACTTTTCTTTGAAGG
ATGACTACAGACATTAAATAATCGAAGTAATTTGGGTGGGGAGAAGAGGCAGATTCAATTTCTTTAACCAGTCTGAAGTTTCATTATGATACAAAAGA
AGATGAAAATGGAAGTGGAATATAAGGGGATGAGGAAGGCATGCCTGGACAAACCCCTCTTTTAAAGATGTGCTTCAATTTGTATAAATGGTGTTTTC
ATGTAATAAATACATCTTGGAGGAGC
```

### Code

```
Base.include(Main, "SymGrpAndReps.jl")
using GRUtils
using StatsBase
using Statistics

obs = readlines("UPENNPLASMA.csv")
VCmap =
Dict{String,Int8}{"bl"=>0,"m06"=>6,"m12"=>12,"m18"=>18,"m24"=>24,"m36"=>36,"m48"
=>48})
parsedObs = map(obs[2:end]) do line
    l = split(replace(line, '"' => " ", " ,"), 1:4)
    return (parse(Int16,l[1]), VCmap[l[2]], l[3]==" " ? -1 : parse(Float64,l[3]), l[4]==" " ? -1 :
parse(Float64,l[4]))
end
cleanObs = filter(y-> !(-1 in y), parsedObs)
RIDs = sort(unique(first.(cleanObs)))
N = length(RIDs)

byPat = []
for r in RIDs
    patObs = filter(x->x[1]==r,cleanObs)
    if length(patObs)!=1
        t = map(x->x[2],patObs)
        AB40 = map(x->x[3],patObs)
        AB42 = map(x->x[4],patObs)
        push!(byPat, (r,t,AB40,AB42,AB42 ./ AB40))
    end
end
Samp = filter(p->p[2]==[0,12,24,36],byPat)
n = length(Samp)
sRIDs = [Int(p[1]) for p in Samp]

DX = Dict{Int,Array{Int}}{r=>[] for r in sRIDs}
info = readlines("DXSUM_PDXCONV_ADNIALI.csv")
for line in info[2:end]
    l = split(replace(line, '"' => " ", " ,"), 1:4)
    r = parse(Int,l[3])
    if r in sRIDs
        if length(l[12])!=0
            d = parse(Int,l[12])
            push!(DX[r], d)
        end
    end
end
```

```

end
end
end

for r in sRIDs DX[r] = unique(DX[r]) end

# 3=AD,2=MCI,1=NL
G = zeros((3,4,372,103));
Gr = zeros((3,4,134));
D = zeros((2,2,2,4,372,103));
Dr = zeros((2,2,2,4,134));
for p in Samp
    c = length(DX[p[1]])
    e = Int.(DX[p[1]])
    e_1 = Int(1 in e)+1
    e_2 = Int(2 in e)+1
    e_3 = Int(3 in e)+1
    g_k = maximum(e)
    for i in 1:4
        D[e_1, e_2, e_3, i, Int(round(p[3][i])), Int(round(p[4][i])) ] += 1/4
        Dr[e_1, e_2, e_3, i, Int(round(p[4][i]/p[3][i] *100)) ] += 1/4
        G[ g_k, i,Int(round(p[3][i])), Int(round(p[4][i]))] += 1/4
        Gr[ g_k, i,Int(round(p[4][i]/p[3][i] *100)) ] += 1/4
    end
end
end
D/=n;
Dr/=n;
G/= n;
Gr/=n;

```

```

d = zeros(2,2,2,3,372,103);
dr = zeros(2,2,2,3,134);
for i in 1:3
    d[:, :, :, i, :] = (D[:, :, :, i+1, :] - D[:, :, :, i, :])
    dr[:, :, :, i, :] = (Dr[:, :, :, i+1, :] - Dr[:, :, :, i, :])
end

```

$F(x::\text{Array}) = [\text{sum}([x[j] * \exp(-im * 2 * \pi * k * (j-1) / \text{length}(x)) \text{ for } j \text{ in } 1:\text{length}(x)]) \text{ for } k \text{ in } 1:\text{length}(x)]$   
 $\otimes(A::\text{Array}\{T\}, B::\text{Array}\{T\}) \text{ where } T <: \text{Number} = \text{prod}(\text{Base.product}(A, B))$   
 $E(X::\text{Array}, i::K) \text{ where } \{K <: \text{Integer}\} = \text{dropdims}(\text{sum}(X, \text{dims}=\text{setdiff}(1:\text{ndims}(X), i)), \text{dims}=\text{tuple}(\text{setdiff}(1:\text{ndims}(X), i) \dots))$



```

E(X::Array,I::NTuple)
=dropdims(sum(X,dims=setdiff(1:ndims(X),I)),dims=tuple(setdiff(1:ndims(X),I)...))
cov(X::Array,i::K,j::K) where {K<:Integer} = E(X,(i,j))- E(X,i)⊗E(X,j) / sum(E(X,i)⊗E(X,j))
cov(X::Array,I::NTuple,j::K) where {K<:Integer} = E(X,(I...,j))-E(X,I)⊗E(X,j)
cov(X::Array,i::K,J::NTuple) where {K<:Integer} = E(X,(i,J...))-E(X,i)⊗E(X,J)
cov(X::Array,I::NTuple,J::NTuple) =E(X,(I...,J...))-E(I)⊗E(J)

```

```

data = (t=[p[2] for p in Samp],AB40=[p[3] for p in Samp],AB42=[p[4] for p in
Samp],ABratio=[p[5] for p in Samp])

```

```

U = sum([1/6 * permutedims(D,vcat(p,[4,5,6])) for p in Sym(3)]);
A = D-U;

```

```

c = E(G,1);
G_u1 = sum([G[i,:,:] for i in 1:3]);
G_a = sum([G[i,:,:]- c[i]*G_u1 for i in 1:3]);
k = sum(1/2*abs.(G_a))
1/ (1+k)

```

```

Gr_u1 = sum([Gr[i,:,:] for i in 1:3]);
Gr_a = sum([Gr[i,:,:]- c[i]*Gr_u1 for i in 1:3]);
k_r = sum(1/2*abs.(Gr_a))
1/ (1+k_r)

```

```

Alz = G[3,:,:];
Alz_ut = sum([Alz[t,:,:] for t in 1:4]);
Alz_a = sum([Alz[t,:,:] - Alz_ut for t in 1:4]);
k_alz = sum(1/2*abs.(Alz_a))
1/ (1+k_alz)

```

```

Alzr = Gr[1,:,:];
Alzr_ut = sum([Alzr[t,:] for t in 1:4]);
Alzr_a = sum([Alzr[t,:] - Alzr_ut for t in 1:4]);
k_alzr = sum(1/2*abs.(Alzr_a))
1/ (1+k_alzr)

```

```

Alz = G[3,:,:];
Alz_ut = sum([Alz[t,:,:] for t in 1:4]);
Alz_a = sum([Alz[t,:,:] - Alz_ut for t in 1:4]);
k_alz = sum(1/2*abs.(Alz_a))
1/ (1+k_alz)

```