Unsupervised Learning

- Interaktion mellem β-cyclodextriner og kolesterol -

Speciale - Forår 2019

Amanda - Sofie Bang Kronborg

Aalborg Universitet
Institut for Matematiske Fag

AALBORG UNIVERSITET
STUDENTERRAPPORT

Institut for Matematiske Fag
Skjernvej 4A
9220 Aalborg Ø
http://math.aau.dk

Titel:
Unsupervised Learning

Tema:
Interaktion mellem β-cyclodextriner og kolesterol

Projektperiode:
Speciale, forårssemesteret 2019

Studerende:
Amanda - Sofie Bang Kronborg

Vejleder:
Poul Svante Eriksen

Oplagstal: 1

Sidetal: 29

Afleveringsdato:
6. juni 2019

Abstract:

In recent years the interaction between β-cyclodextrin and cholesterol has been an area of much research. This is mainly due to the assumption that β-cyclodextrin has a positive effect on cholesterol based diseases.
In this project different unsupervised learning tools are used to study the interaction between the two molecules, β-cyclodextrin and cholesterol.
The examined data set consist of computer simulations of the above stated interaction, which provides 1001 different images.
It is desired to apply different clustering methods to achieve a representation of the data set with a reduced number of images. This is shown to be feasible.

Indholdsfortegnelse

Forord

Dette speciale er udarbejdet på forårssemesteret 2019 ved Institut for Matematiske Fag, Aalborg Universitet af Amanda - Sofie Bang Kronborg. Det overordnede statistiske emne i projektet er *unsupervised learning*, hvorfra der anvendes redskaber til at undersøge computersimuleret data af interaktionen mellem β-cyclodextrin og kolesterol.

I projektet præsenteres først relevant teori, som herefter anvendes til at analysere datasættet. Det forventes at læseren har en god forståelse for sandsynlighedsregning og statistik.

Referencer forefindes i begyndelsen af kapitler og afsnit, hvis disse afviger fra resten af kapitlet, og betegnes med tal, for eksempel [tal], med en tilhørende kilde i bibliografien. Supplement i form af appendiks kan findes sidst i projektet, og relevante referencer hertil findes undervejs i projektet.

Datasættet analyseres i programmet R, som er et software environment til statistiske beregninger. Figurer uden referencer er udarbejdet ved brug af R.

Jeg vil gerne takke min vejleder Poul Svante Eriksen for konstruktiv kritik og god vejledning undervejs i forløbet. Jeg vil ydermere takke samarbejdspartner Casper Steinmann Svendsen fra Institut for Kemi og Biovidenskab for inspiration til projektet og for data.

Aalborg Universitet, 6. juni 2019

Amanda - Sofie Bang Kronborg
<akronb14@student.aau.dk>

Indledning

Indledningen er baseret på [1].

På Institut for Kemi og Biovidenskab arbejdes med at forstå interaktionen mellem kolesterol og β-cyclodextriner, da man har en formodning om, at denne interaktion er grund til de positive effekter, β-cyclodextriner har på kolesterol-baserede sygdomme. Blandt andet har det vist sig igennem undersøgelser, at β-cyclodextriner har virket nedsættende på progressionen af Niemann-Pick C1 sygdommen.

Cyclodextriner har været brugt i lægemidler på grund af deres hydrofobe indre og hydrofile ydre, som muliggør kompleksdannelse med hydrofobe forbindelser. Det har sidenhen vist sig at være mere kompliceret end dette, og det er derfor nødvendigt at få udviklet nye metoder til at undersøge interaktionen.

På Institut for Kemi og Biovidenskab undersøges disse problemstillinger, hvor der både foretages forsøg i laboratoriet, samtidig med at der udarbejdes computersimulationer af vekselvirkninger mellem kolesterol og β-cyclodextrin. Datasættet, som produceres igennem simulationerne, indeholder imidlertid mange datapunkter, og det er derfor eftertragtet at kunne reducere datasættet til den efterfølgende analyse, således at det repræsenterer ekstreme og hyppigt forekommende tilstande.

Dette projekt vil anvende computersimulationerne af ovenstående problem med henblik på at undersøge de underliggende strukturer i datasættet, som vil blive anvendt til at reducere datasættet.
For at kunne behandle datasættet og opnå en reduktion af dette, arbejdes der i dette projekt med det overordnede emne, *unsupervised learning*. Herunder anvendes principal komponent analyse til forbehandling af datasættet og klyngeanalyse bruges til at analysere de computersimulerede data.

Kapitel 1

Unsupervised Learning

Dette kapitel er baseret på [3] og [4].

Unsupervised learning er en samling af statistiske værktøjer til den type af problemer, hvor der til et givet datasæt, $X_1, X_2, \ldots, X_p$, målt på n observationer ikke haves en tilhørende responsvariabel, Y. Målet med analysen er at opdage interessante ting ved målingerne, så som en informativ visuel repræsentation, eller undergrupper blandt variablene eller observationerne.

I dette kapitel gennemgås to metoder inden for *unsupervised learning*, nemlig *klyngeanalyse* og *principal komponent analyse*, forkortet PCA. Både klyngedannelse og PCA har til formål at simplificere datasættet. Forskellen mellem de to er dog, at PCA forsøger at finde en lavdimensionel repræsentation af observationerne, hvor målet er, at denne repræsentation forklarer størstedelen af variansen. Klyngedannelse har derimod til formål at repræsentere datasættet i undergrupper blandt observationerne.

1.1 Klyngeanalyse

Klyngeanalyse er en metode, der kan anvendes på et datasæt for at finde undergrupper, såkaldte klynger, i datasættet. Det ønskes, at disse klynger adskiller observationerne i datasættet, således at observationer inden for hver klynge er mere nært beslægtede med hinanden end med observationer, som er tildelt andre klynger. Klyngeanalyse er et problem inden for *unsupervised learning*, fordi der arbejdes med at finde strukturer, i dette tilfælde klynger, på grundlag af datasættet.

I de følgende afsnit gennemgås tre metoder inden for klyngeanalyse kaldet *K-middel klyngedannelse*, *hierarkisk klyngedannelse* og *mixture modeling*.
Ved K-middel klyngedannelse ønskes det at opdele observationerne i et forudbe-

stemt antal klynger. Observationerne inddeles først tilfældigt, hvorefter de tildeles på ny, til den klynge, hvis middel, den er tættest relateret til. Dette gentages, til den mest optimale klyngetildeling er opnået.
Ved hierarkisk klyngedannelse ønskes det at inddele klyngerne i et naturligt hierarki. Dette indebærer at gruppere klyngerne selv, således at klynger inden for samme gruppe på hvert niveau af hierarkiet ligner hinanden mere end dem i andre grupper på samme niveau. Ved denne metode opnås da en træ-lignende repræsentation af observationerne, kaldet et *dendrogram*.
Ved *mixture modeling* ønskes det at opdele observationerne i et bestemt antal klynger, som ved K-middel klyngedannelse, men i stedet for at punkterne tildeles deterministisk, tildeles de probabilistisk til klyngerne.

For at klyngedannelsen giver mening, skal klyngerne opfylde følgende egenskaber. Lad $C_1, \ldots, C_K$ betegne mængder, som indeholder indekserne af de observationer, hver klynge indeholder. For mængderne gælder da, at:

- $C_1 \cup C_2 \cup \cdots \cup C_K = \{1, \ldots, n\}$. Med andre ord, så er hver observation indeholdt i mindst én af de K klynger.

- $C_k \cap C_{k'} = \emptyset$ for alle $k \neq k'$. Med andre ord, så overlapper klyngerne ikke, altså er hver observation kun tildelt én klynge.

1.1.1 Mål for forskellighed

For at et datasæt kan inddeles i klynger, skal der defineres et mål, som angiver, hvor forskellige observationerne er. Et sådan mål kaldes et *mål for forskellighed*. Valget af dette mål er fundamentalt for alle metoder inden for klyngeanalyse og afhænger ofte af applikationen. Det er derfor vigtigt at basere denne på viden omkring datasættet, der undersøges.
Et eksempel på et mål for forskellighed, som ofte anvendes, er den euklidiske afstand mellem to observationer, men også korrelation bruges som et mål. Eftersom at klyngeanalyse grupperer observationerne ud fra definitionen på målet, afhænger den resulterende klyngedannelse kraftigt af dette valg. Det er derfor vigtigt at overveje hvilket mål, der bedst forklarer forskellighederne.

Et datasæt er sommetider repræsenteret direkte, beskrevet ved forskelle eller ligheder mellem objekterne i datasættet. Denne type datasæt kan beskrives ved hjælp af en $N \times N$ matrix, $\mathbb{D}$, hvor N er antallet af objekter, og $d_{ii'}$ beskriver forskellen mellem objekterne i og i'. I dette tilfælde kan denne matrix benyttes direkte som input til klyngedannelses-algoritmen.
Oftest ses det dog, at datasættet indholder målinger, x_{ij} af de $i = 1, 2, \ldots, N$ objekter målt på $j = 1, 2, \ldots, p$ forskellige egenskaber. Der skal altså konstrueres et mål

for forskellighed mellem de forskellige objekter. Normalt vælges et mål mellem værdierne af den j'te egenskab, $d_j(x_{ij}, x_{i'j})$, og da defineres

$$D(x_i, x_{i'}) = \sum_{j=1}^{p} d_j(x_{ij}, x_{i'j}) \tag{1.1}$$

som forskellen mellem objekterne i og i'.

1.2 Klyngedannelses-algoritmer

Klyngedannelses-algoritmer kan inddeles i tre adskilte typer: kombinatoriske algoritmer, *mixture modeling* og *mode seeking*. I dette projekt gennemgås kun de to første typer.
Kombinatoriske algoritmer arbejder direkte med det observerede data uden reference til en underliggende sandsynlighedsmodel.
Mixture modeling antager, at datasættet er en i.i.d stikprøve fra en population, som kan beskrives ved hjælp af en tæthedsfunktion. Denne tæthedsfunktion er karakteriseret ved en parametriseret model, som består af en blanding af komponent-tæthedsfunktioner, hvor hver af disse beskriver en klynge. Modellen fittes da til datasættet ved brug af maksimum likelihood estimering.

1.2.1 Kombinatoriske algoritmer

Ved kombinatoriske algoritmer betegnes hver observationer entydigt med et heltal, $i \in \{1, \ldots, N\}$. Et præspecificeret antal af klynger vælges, og hver klynge betegnes med et heltal, $k \in \{1, \ldots, K\}$, hvor $K < N$. Hver observation tildeles da én og kun én klynge ved hjælp af en *encoder*, $k = C(i)$, som tildeler den i'te observation til den k'te klynge. Det ønskes da at finde den speficikke *encoder*, $C^*(i)$, som opnår det ønskede mål. Tildelingerne af de N observationer justeres af en såkaldt tabsfunktion, som indikerer i hvilken grad, klyngedannelsen ikke er opnået. For at opnå det ønskede mål skal denne tabsfunktion altså minimeres. En tilgang til dette er at specificere denne tabsfunktion matematisk, for derefter at forsøge at minimere den ved hjælp af en kombinatorisk algoritme. Da formålet er at tildele lignende punkter samme klynge, vil en intuitiv tabsfunktions være

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} d(x_i, x_{i'}). \tag{1.2}$$

Dette opstiller et kriterie for, hvorvidt observationer, som er tildelt den samme klynge, er tætte på hinanden. Denne refereres ofte til som *within-cluster point scatter*.

Den totale *point scatter*,

$$T = \frac{1}{2}\sum_{i=1}^{N}\sum_{i'=1}^{N} d_{ii'} = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\left(\sum_{C(i')=k} d_{ii'} + \sum_{C(i')\neq k} d_{ii'}\right), \tag{1.3}$$

hvor $d_{ii'} = d(x_i, x_{i'})$, kan dekomponeres som $T = W(C) + B(C)$. Denne afhænger ikke af klyngetildelingerne, og er altså konstant, givet et datasæt.

$$B(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(i')\neq k} d_{ii'} \tag{1.4}$$

er *between-cluster point scatter*, som antager en stor værdi, når observationer fra forskellige klynger er meget forskellige. Det gælder altså, at $W(C) = T - B(C)$, og at minimere $W(C)$ er ækvivalent med at maksimere $B(C)$.

K-middel klyngedannelse

K-middel klyngedannelse er en nem metode til at opdele et datasæt i K forskellige, ikke-overlappende klynger. Som nævnt, angives på forhånd antallet af klynger, som datasættet skal inddeles i. Da tildeler K-middel klyngedannelses-algoritmen hver observation til en af de K klynger.
Det ønskes at opdele observationerne i K klynger, sådan at $W(C)$ minimeres. For at det er muligt at løse dette, skal målet for forskellighed defineres. Oftest anvendes den kvadrerede euklidiske afstand,

$$d(x_i, x_{i'}) = \sum_{j=1}^{p}(x_{ij} - x_{i'j})^2 = ||x_i - x_{i'}||^2, \tag{1.5}$$

som mål for forskellighed, hvilket giver følgende *within-cluster point scatter*,

$$W(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(i')=k} ||x_i - x_{i'}||^2 \tag{1.6}$$

$$= \sum_{k=1}^{K} N_k \sum_{C(i)=k} ||x_i - \bar{x}_k||^2, \tag{1.7}$$

hvor $\bar{x}_k = (\bar{x}_{1k}, \ldots, \bar{x}_{pk})$ er middelvektoren af den k'te klynge og $N_k = \sum_{i=1}^{K} I(C(i) = k)$. Kriteriet minimeres altså ved at tildele de N observationer til de K klynger, sådan at den gennemsnitlige forskellighed af observationerne inden for en klynge minimeres i forhold til klyngens middelværdi.
Bemærk, at for enhver mængde af observationer, S, gælder

$$\bar{x}_s = \arg\min_m \sum_{i\in S} ||x_i - m||^2. \tag{1.8}$$

En algoritme, som løser

$$C^* = \min_C \sum_{k=1}^{K} N_k \sum_{C(i)=k} ||x_i - \bar{x}_k||^2, \tag{1.9}$$

skal altså løse følgende optimeringsproblem

$$\min_{C,\{m_k\}_1^K} \sum_{k=1}^{K} N_k \sum_{C(i)=k} ||x_i - m_k||^2. \tag{1.10}$$

En algoritme, som løser dette, er givet i følgende.

Algoritme 1.1 (K-middel klyngedannelse)

1. For en given klynge, C, minimeres ligning (1.10), svarende til den totale varians i klyngen, i forhold til $\{m_1, \ldots, m_K\}$, hvilket giver middelværdien af de nuværende tildelte klynger.

2. Givet den nuværende mængde af middelværdier, $\{m_1, \ldots, m_K\}$, minimeres ligning (1.10) ved at tildele hver observation til den klynge, der er tættest på. Altså

$$C(i) = \arg \min_{1 \leq k \leq K} ||x_i - m_k||^2. \tag{1.11}$$

3. Ovenstående gentages indtil tildelingerne ikke ændres.

Algoritmen finder kun et lokalt og ikke et globalt optimum, som afhænger af den oprindelige inddeling i klynger. Det er derfor vigtigt at køre algoritme flere gange med flere forskellige oprindelige inddelinger. Da kan den bedste løsning vælges, altså hvor $W(C)$ er mindst.

Hierarkisk klyngedannelse

Hierarkisk klyngedannelse er en metode, som ikke kræver et forudbestemt antal af klynger, men da algoritmen fusionerer eller splitter klynger skal et yderligere mål defineres, nemlig et mål for forskellighed mellem grupper af observationer.
Som navnet antyder, så resulterer klyngedannelsen i en hierarkisk repræsentation. Ved det nederste niveau af hierarkiet indeholder klyngerne kun én enkelt observation, og ved det øverste niveau findes kun én klynge, som indeholder alle observationerne. Dette repræsenteres i et såkaldt dendrogram.

Hvert niveau af hierarkiet repræsenterer en gruppering af datasættet i disjunkte klynger af observationerne. Overordnet er hierarkiet en repræsentation af en ordnet sekvens af disse grupperinger. Det er således op til brugeren at beslutte, hvilket niveau der er den bedste repræsentation af en "naturlig"klyngedannelse, forstået på den måde at observationerne indenfor denne klynge har mere til fælles med hinanden end observationerne i andre klynger ved det niveau.

Det endelige antal af klynger vælges ved at lave et horisontalt snit på tværs af dendrogrammet. Højden af snittet har samme rolle som valget af K i K-middel klyngedannelse, det styrer mængden af klynger, som dannes. Altså kan et enkelt dendrogram benyttes til at opnå ethvert antal af klynger.

Hierarkisk klyngedannelse kan gøres på to måder, enten *agglomerativ* eller *splittende*. Agglomerativ klyngedannelse starter fra bunden og fusionerer to klynger til en ved hvert niveau. Splittende klyngedannelse starter fra toppen og deler en klynge til to ved hvert niveau.

Agglomerativ klyngedannelse

Den agglomerative klyngedannelse starter med, at hver enkelt observation svarer til en klynge. I hvert af de $N-1$ skridt af algoritmen fusioneres de to klynger, som er mindst forskellige. Det er altså nødvendigt at definere, hvad forskellen mellem to klynger er.
Konceptet om forskellen mellem et par af observationer skal altså udvides til forskellen mellem et par af grupper af observationer. Denne udvidelse opnås ved *linkage*. De tre hyppigst brugte typer af *linkage* er *complete, group average* og *single linkage*. Lad G og H være to klynger, som indeholder en eller flere observationer.

- *Single linkage*, SL: den minimale forskel mellem observationerne i klyngerne anvendes som forskellen mellem to klynger, altså

$$d_{SL}(G,H) = \min_{i\in G, i'\in H} d_{ii'}. \tag{1.12}$$

- *Complete linkage*, CL: den maksimale forskel mellem observationerne i klyngerne anvendes som forskellen mellem to klynger, altså

$$d_{CL}(G,H) = \max_{i\in G, i'\in H} d_{ii'}. \tag{1.13}$$

- *Group average linkage*, GA: den gennemsnitlige forskel mellem observationerne i klyngerne anvendes som forskellen mellem to klynger.

$$d_{GA}(G,H) = \frac{1}{N_G \dot{N}_H} \sum_{i\in G} \sum_{i'\in H} d_{ii'}, \tag{1.14}$$

  hvor N_G og N_H er antallet af observationer i de respektive klynger.

Typisk afhænger dendrogrammer meget af valget af linkage.

Splittende klyngedannelse

Den splittende klyngedannelse starter med, at hele datasættet svarer til én klynge. Ved hvert af de $N-1$ skridt af algoritmen splittes én af de eksisterende klynger da i to.
En algoritme for splittende klyngedannelse starter med at tildele alle observationer til en enkelt klynge, G. Da vælges den observation, hvis gennemsnitlige forskel fra alle andre observationer er størst. Denne observation tildeles som den første til en anden klynge, H. Ved hvert efterfølgende skridt overføres den observation, hvis gennemsnitlige afstand fra observationerne i H minus den gennemsnitlige afstand fra observationerne i G er størst, til klyngen H. Dette forsættes til værdien bliver negativ, altså at der ikke er flere observationer i G, som gennemsnitligt er tættere på observationerne i H. Disse to klynger repræsenterer da det andet niveau i hierarkiet.
Hvert efterfølgende niveau produceres ved at gøre samme procedure ved én af de eksisterende klynger. For at vælge hvilken klynge, der skal splittes, ses enten på diameteren af klyngerne, $D_G = \max_{i \in G, i' \in G} d_{ii'}$, eller den største gennemsnitlige forskellighed blandt medlemmer af de forskellige klynger, $\bar{d}_G = \frac{1}{N_G} \sum_{i \in G} \sum_{i' \in G} d_{ii'}$. Opsplitningen fortsættes til alle klynger kun indeholder én observation, eller hvis alle observationer i hver klynge ikke er forskellige fra sine klyngemedlemmer.

1.2.2 Mixture modeling

Dette afsnit er baseret på [2] og [3].

Som nævnt tidligere antages der for de kombinatoriske algoritmer, at der ingen underliggende sandsynlighedsmodel er for datasættet.
Mixture modeling er en metode, som kan anvendes, hvis man derimod antager, at en sådan underliggende model eksisterer. Som navnet antyder, er metoden især brugbar til at modellere data, som udviser en tendens til at have flere underpopulationer. Dette gøres ved at karakterisere tæthedsfunktionen ved en blanding af flere komponent-tæthedsfunktioner, som hver beskriver en klynge af datasættet. Det antages altså, at hvert datapunkt tilhører en af disse underpopulationer, og det forsøges da at estimere en fordeling for hver af disse komponenter. Metoden er da et brugbart redskab til at estimere de tæthedsfunktioner, som beskriver denne model. Det kræves ikke, at det på forhånd vides, hvilke datapunkter der tilhører hvilke underpopulationer, hvilket gør, at problemet også tilhører *unsupervised learning*.

Et eksempel på en mixture model er den *Gaussiske mixture model*. Generelt kan enhver tæthedsfunktion bruges for komponenterne i en mixture model, men den Gaussiske mixture model er den mest anvendte model. Denne er en probabilistisk model, som repræsenterer normalfordelte underpopulationer indenfor en overordnet population, og er givet ved

$$f(x_1, \ldots, x_n) = \prod_{i=1}^{n} \sum_{k=1}^{G} \tau_k \phi_k(\mathbf{x}_i \mid \mu_k, \Sigma_k), \tag{1.15}$$

hvor $\mathbf{x}$ er datasættet, G er antallet af komponenter, τ_k er sandsynligheden for, at en observation tilhører den k'te komponent, hvorom det gælder at $\tau_k \geq 0$ og $\sum_{k=1}^{G} \tau_k = 1$, og

$$\phi_k(\mathbf{x} \mid \mu_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^G |\Sigma_k|}} \exp\left(-\frac{1}{2} (\mathbf{x}_i - \mu_k)^T \Sigma_k^{-1} (\mathbf{x}_i - \mu_k) \right). \tag{1.16}$$

De k komponenter har alle form som ellipsoider med centrum i middelværdierne, μ_k, hvor ellipsoideformene er bestemt af kovariansmatricerne, Σ_k.
Hver kovariansmatrix er parametriseret ved egenværdi-dekompositionen på formen $\Sigma_k = \lambda_k D_k A_k D_k^T$, hvor D_k er en ortogonalmatrix bestående af egenvektorerne, A_k er en diagonalmatrix, hvis elementer er proportionale med egenværdierne af Σ_k, og λ_k er en skalar. Orienteringen af principal komponenterne af Σ_k bestemmes ud fra D_k, mens A_k bestemmer formen for densitetskonturerne. Skalaren λ_k specificerer volumen af den tilsvarende ellipsoide, som er proportional med $\lambda_k^d |A_k|$, hvor d er datasættets dimension.

De forskellige karakteristikker ved fordelingerne, så som orietentering, volumen og form, estimeres som regel fra datasættet. Disse kan enten variere mellem klyngerne eller være ens for alle klynger. Hvis datasættet kun består af én dimension, er der kun to forskellige modeller, som betegnes ved E for ens varians eller V for variabel varians. Hvis datasættet består af mere end en dimension, bestemmer model-identifikationen geometriske karakteristikker ved modellen. De mulige modeller ved brug af `mclust` og deres geometriske karakteristikker er vist i tabel 1.1.

Identifikation	Model	Fordeling	Volumen	Form	Orientering
E	-	-	ens	-	-
V	-	-	variabel	-	-
EII	λI	sfærisk	ens	ens	NA
VII	$\lambda_k I$	sfærisk	variabel	ens	NA
EEI	λA	diagonal	ens	ens	koordinat-akser
VEI	$\lambda_k A$	diagonal	variabel	ens	koordinat-akser
EVI	λA_k	diagonal	ens	variabel	koordinat-akser
VVI	$\lambda_k A_k$	diagonal	variabel	variabel	koordinat-akser
EEE	λDAD^T	ellipsoidisk	ens	ens	ens
EEV	$\lambda D_k A D_k^T$	ellipsoidisk	ens	ens	variabel
VEV	$\lambda_k D_k A D_k^T$	ellipsoidisk	variabel	ens	variabel
VVV	$\lambda_k D_k A_k D_k^T$	ellipsoidisk	variabel	variabel	variabel

Tabel 1.1: Tilgængelige parametriseringer af kovariansmatricen i pakken `mclust` ved brug af EM-algoritmen og deres geometriske karakteristikker.

EM-algoritmen

EM-algoritmen er et populært værktøj til at simplificere besværlige problemer med maksimum likelihood estimering. Navnet er en forkortelse af de to ord *expectation* og *maximization*, som er de to skridt algoritmen udfører iterativt.
I det følgende tages udgangspunkt i et eksempel, hvor datasættet modelleres med en Gaussisk *mixture model* med to komponenter. Altså modelleres Y som en blanding af to normalfordelinger,

$$\begin{aligned} Y_1 &\sim N(\mu_1 \sigma_1^2), \\ Y_2 &\sim N(\mu_2 \sigma_2^2), \\ Y &= (1-\Delta) \cdot Y_1 + \Delta \cdot Y_2, \end{aligned} \tag{1.17}$$

hvor $\Delta \in \{0,1\}$ med $\Pr(\Delta = 1) = \pi$.
Lad $\phi_\theta(x)$ betegne normalfordelingen med parametrene $\theta = (\mu, \sigma^2)$, da er densiteten af Y givet ved

$$g_Y(y) = (1-\pi)\phi_{\theta_1}(y) + \pi\phi_{\theta_2}(y). \tag{1.18}$$

Log-likelihood-funktionen er da

$$\begin{aligned}\ell(\theta;\mathbf{y}) &= \log\left(\prod_{i=1}^{N} g_Y(y_i)\right)\\ \ell(\theta;\mathbf{y}) &= \log\left(\prod_{i=1}^{N}(1-\pi)\phi_{\theta_1}(y_i)+\pi\phi_{\theta_2}(y_i)\right) \qquad (1.19)\\ \ell(\theta;\mathbf{y}) &= \sum_{i=1}^{N}\log[(1-\pi)\phi_{\theta_1}(y_i)+\pi\phi_{\theta_2}(y_i)].\end{aligned}$$

Der findes en mere simpel tilgang end maksimering af denne, nemlig en procedure, hvor det i stedet antages, at der eksisterer ikke-observerede latente variable, Δ_i, som kan antage værdierne 0 og 1. Hvis værdierne af disse latente variable er kendt, så er log-likelihood-funktionen givet ved

$$\ell(\theta;\mathbf{y},\delta) = \sum_{i=1}^{N}[(1-\delta_i)\log\phi_{\theta_1}(y_i)+\delta_i\log\phi_{\theta_2}(y_i)] \qquad (1.20)$$

$$+\sum_{i=1}^{N}[(1-\delta_i)\log(1-\pi)+\delta_i\log\pi], \qquad (1.21)$$

og maksimum likelihood estimaterne af μ_1 og σ_1^2 er da middelværdien og variansen af den stikprøve, hvor $\Delta_i = 0$, ligeledes for μ_2 og σ_2^2, hvor $\Delta_i = 1$. Estimatet af π er proportionen af $\Delta_i = 1$.

Eftersom at værdierne af Δ_i ikke er kendt, erstattes disse med deres forventede værdier givet de observerede data og parametrene,

$$\gamma_i(\theta) = E(\Delta_i \mid \theta, \mathbf{Z}) = \Pr(\Delta_i = 1 \mid \theta, \mathbf{Z}), \qquad (1.22)$$

som også kaldes *ansvaret* for model 2 for den i'te observation. For Gaussiske *mixture models* med to komponenter bruges EM-algoritmen som ses i algoritme 1.2. Denne kan generaliseres til multivariate Gaussiske *mixture models* med flere end to komponenter.

I *expectation*-skridtet laves en blød tildeling af observationerne til en af de to modeller ved brug af de nuværende estimater af parametrene. I *maximization*-skridtet, bruges de udregnede ansvar til at lave en vægtet opdatering af estimaterne af parametrene.
De initielle værdier for de forskellige parametre kan vælges på følgende måde: Middelværdierne, $\hat{\mu}_1$ og $\hat{\mu}_2$, kan vælges tilfældigt som to af observationerne, y_i. Varianserne, $\hat{\sigma}_1^2$ og $\hat{\sigma}_2^2$, kan vælges som variansen af stikprøven, $\sum_{i=1}^{N}\frac{(y_i-\bar{y})^2}{N}$. Sandsynligheden, $\hat{\pi}$, kan vælges til at starte med en værdi på 0,5.

Algoritme 1.2 (EM-algoritmen)

1. Vælg initial-værdier for parametrene $\hat{\mu}_1, \hat{\sigma}_1^2, \hat{\mu}_2, \hat{\sigma}_2^2, \hat{\pi}$.
2. *Expectation*: udregn ansvarene,

$$\hat{\gamma}_i = \frac{\hat{\pi}\phi_{\hat{\theta}_2}(y_i)}{(1-\hat{\pi})\phi_{\hat{\theta}_1}(y_i) + \hat{\pi}\phi_{\hat{\theta}_2}(y_i)}, \; i = 1, 2, \ldots, N. \tag{1.23}$$

3. *Maximization*: beregn de vægtede middelværdier og varianser,

$$\hat{\mu}_1 = \frac{\sum_{i=1}^N (1-\hat{\gamma}_i) y_i}{\sum_{i=1}^N (1-\hat{\gamma}_i)} \tag{1.24}$$

$$\hat{\sigma}_1^2 = \frac{\sum_{i=1}^N (1-\hat{\gamma}_i)(y_i - \hat{\mu}_1)^2}{\sum_{i=1}^N (1-\hat{\gamma}_i)} \tag{1.25}$$

$$\hat{\mu}_2 = \frac{\sum_{i=1}^N \hat{\gamma}_i y_i}{\sum_{i=1}^N \hat{\gamma}_i} \tag{1.26}$$

$$\hat{\sigma}_2^2 = \frac{\sum_{i=1}^N \hat{\gamma}_i (y_i - \hat{\mu}_2)^2}{\sum_{i=1}^N \hat{\gamma}_i}, \tag{1.27}$$

   og sandsynligheden, $\hat{\pi} = \sum_{i=1}^N \frac{\hat{\gamma}_i}{N}$.

4. Gentag trin 2 og 3 indtil resultaterne konvergerer.

Bayesian Information Criterion

For at den mest optimale model bliver valgt, anvendes *Bayesian Information Criterion*, forkortet BIC, som er defineret ved

$$\text{BIC} \equiv 2 \, \text{loglik}_{\mathcal{M}}(\mathbf{x}, \theta_k^*) - (\#\text{param})_{\mathcal{M}} \log(n), \tag{1.28}$$

hvor $\text{loglik}_{\mathcal{M}}(\mathbf{x}, \theta_k^*)$ er den maksimerede loglikelihood-funktion for modellen og datasættet, $(\#\text{param})_{\mathcal{M}}$ er antallet af uafhængige parametre, der skal estimeres i modellen $\mathcal{M}$, og n er antallet af observationer i datasættet.
Det ønskes at vælge en model, som både har en høj likelihood men som samtidig er simpel med få parametre, altså vælges den model, som har den største BIC.

1.3 Principal komponent analyse

Dette afsnit er baseret på [4].
PCA tilhører *unsupervised learning*, eftersom det kun involverer en mængde af *features* $X_1, X_2, \ldots, X_p$, uden en associeret respons, Y. PCA finder en lav-dimensional repræsentation af datasættet, som indeholder så meget information om variationen som muligt. Ideen er, at de n observationer er i et p-dimensionalt rum, men ikke alle dimensioner er lige interessante. PCA bruges da til at finde et mindre antal af dimensioner, som er så interessante som muligt. Hver af disse dimensioner er en linearkombination af de p *features*.

1.3.1 Principal komponenter

Den første principal komponent af $X_1, \ldots, X_p$ er den normaliserede linearkombination af disse,

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \cdots + \phi_{p1}X_p,$$

som har den største varians. Ved normaliseret menes, at $\sum_{j=1}^{p} \phi_{j1}^2 = 1$. Elementerne $\phi_{11}, \ldots, \phi_{p1}$ kaldes for *loadings* af den første principal komponent, som sammen udgør principal komponent *loading*-vektoren, $\phi_1 = (\phi_{11}, \phi_{21} \cdots \phi_{p1})^T$. Disse *loadings* begrænses, så deres sum af kvadrater er lig en.

Givet et $n \times p$ datasæt, $\mathbf{X}$, antages at hver variabel i $\mathbf{X}$ er centreret, så de har en middelværdi på nul, eftersom det kun er variansen, der er interessant. Da findes en linearkombination af stikprøven på formen

$$z_{i1} = \phi_{11}x_{i1} + \phi_{21}x_{i2} + \cdots + \phi_{p1}x_{ip}, \tag{1.29}$$

som har den største varians, under begrænsning af at $\sum_{j=1}^{p} \phi_{j1}^2 = 1$.
Altså er den første principal komponent *loading*-vektor en løsning til optimeringsproblemet,

$$\max_{\phi_{11},\ldots,\phi_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1}x_{ij} \right)^2 \right\} \text{ under begrænsning af } \sum_{j=1}^{p} \phi_{j1}^2 = 1, \tag{1.30}$$

hvilket, ud fra ligning 1.29, svarer til

$$\max_{\phi_{11},\ldots,\phi_{p1}} \left\{ \frac{1}{n} \sum_{i=1}^{n} z_{i1}^2 \right\} \text{ under begrænsning af } \sum_{j=1}^{p} \phi_{j1}^2 = 1. \tag{1.31}$$

Eftersom $\frac{1}{n}\sum_{i=1}^{n} x_{ij} = 0$, er gennemsnittet af $z_{11}, \ldots, z_{n1}$ også nul. Altså er det variansen af stikprøven som maksimeres. Fremover betegnes $z_{11}, \ldots, z_{n1}$ som *scores* af den første principal komponent. Optimeringsproblemet kan løses ved hjælp af egenværdi dekomposition.

Geometrisk fortolkning

Der findes en fin geometrisk fortolkning af den første principal komponent, hvor *loadning*-vektoren ϕ_1 definerer en retning i det n-dimensionale rum. I retningen af denne vektor varierer datasættet mest. Ydermere, hvis de n datapunkter, $x_1, \ldots, x_n$, projiceres over på denne vektor, så er værdierne af de projekterede punkter lig principal komponenternes *scores*, $z_{11}, \ldots, z_{n1}$.

Når den første principal komponent, Z_1, er fundet, kan den anden principal komponent, Z_2, bestemmes. Den anden principal komponent er linearkombinationen af $X_1, \ldots, X_p$, som har den maksimale varians af alle de linearkombinationer, som er ukorrelerede med Z_1. Disse *scores* af den anden principal komponent er på formen

$$z_{i2} = \phi_{12}x_{i1} + \phi_{22}x_{i2} + \cdots + \phi_{p2}x_{ip}, \tag{1.32}$$

hvor ϕ_2 er den anden principal komponent *loading*-vektor indeholdende elementerne $\phi_{12}, \phi_{22}, \ldots, \phi_{p2}$. Det viser sig, at begrænsningen ved at Z_2 skal være ukorreleret med Z_1 er ækvivalent med at begrænse ved, at retningen af ϕ_2 skal være vinkelret med retningen af ϕ_1.
For at finde ϕ_2 løses et optimeringsproblem lignende det i ligning (1.30) med en yderligere begrænsning på, at ϕ_2 skal være ortogonal med ϕ_1.

En anden nyttig fortolkning af principal komponenterne er, at komponenterne beskriver lavdimensionelle lineære flader, som er tættest på observationerne. Altså har den første principal komponent *loading*-vektor den meget specielle egenskab, at den er linjen i det p-dimensionelle rum, som er tættest på de n observationer i euklidisk afstand. Fordelen ved denne fortolkning er, at der findes en enkelt dimension af datasættet, som ligger så tæt som muligt på alle datapunkterne, eftersom at sådan en linje, som *loading*-vektoren beskriver, med stor sandsynlighed giver en god opsummering af datasætttet.

Ved denne fortolkning giver de første M principal komponent *score*-vektorer og de første M principal komponent *loading*-vektorer den bedste M-dimensionelle approksimation af den i'te observation, altså $x_{ij} \approx \sum_{m=1}^{M} z_{im}\phi_{ji}$. De kan altså tilsammen give en god approksimation af datasættet, hvis M er tilstrækkeligt stor. Når $M = \min(n-1, p)$ så er repræsentationen eksakt, altså $x_{ij} = \sum_{m=1}^{M} z_{im}\phi_{jm}$.

Skalering af variable

Ud over at PCA algoritmen starter med at centrere variablene, så de har en middelværdi på nul, så afhænger resultatet af PCA også af, om variablene er blevet skaleret individuelt inden analysen. Hvis PCA udføres på ikke-skalerede variable,

så vil en konsekvens være, at den første principal komponent *loading*-vektor har en stor *loading* for den variabel, som har den største varians.
Da det er fordelagtigt at principal komponenterne ikke afhænger af et arbitrært valg af skalering, vælges det ofte at skalere hver variabel til at have en standard afvigelse på en. Det er dog i nogle tilfælde, for eksempel hvis variablene er målt i samme enhed, ikke ønskeligt at skalere variablerne til at have en standardafvigelse på en.

Entydighed af principal komponenter

Hver principal komponent *loading*-vector er entydig op til en ændring af fortegn, da de beskriver en retning i det p-dimensionale rum. Ligeledes er *score*-vektorerne entydige op til en ændring af fortegn, da variansen af Z er den samme som variansen af $-Z$.

Kapitel 2

Dataanalyse

2.1 Beskrivelse af datasæt

I dette projekt arbejdes der med et datasæt, som består af 1001 .pdb-filer. Hver af disse filer indeholder forskellige informationer om de to molekyler, β-cyclodextrin og kolesterol, svarende til en simulation af interaktionen mellem disse over ti nanosekunder.
Følgende informationer om de to molekyler, altså de i alt 221 atomer, anvendes i dataanalysen: atomets type, atomets placering i koordinatsystemet samt hvilket molekyle atomet tilhører. På figur 2.1 ses et plot af de to molekyler fra en tilfældig .pdb-fil.

Figur 2.1: Struktur af β-cyclodextrin og kolesterol. På billedet repræsenterer farverne sort, rød og hvid/grøn henholdsvis carbon, oxygen og hydrogen.

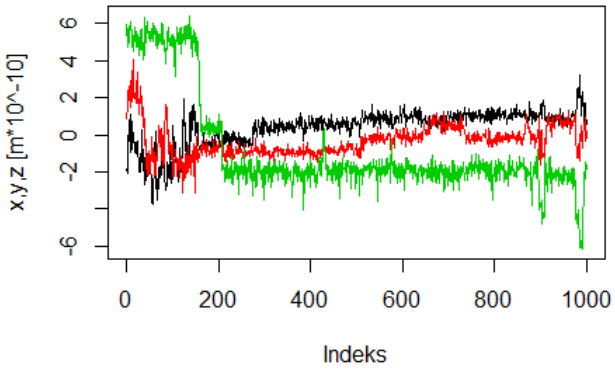
2.2 Indledende dataanalyse

Simulationer af molekylerne viser, hvordan den samlede struktur bevæger sig i forhold til koordinatsystemet. Da det kun er selve interaktionen mellem de to molekyler, der undersøges, ønskes det som indledning til dataanalysen at rotere og translatere billederne, så β-cyclodextrinet er placeret ens i koordinatsystemet i alle filer. Dette gøres ved følgende:

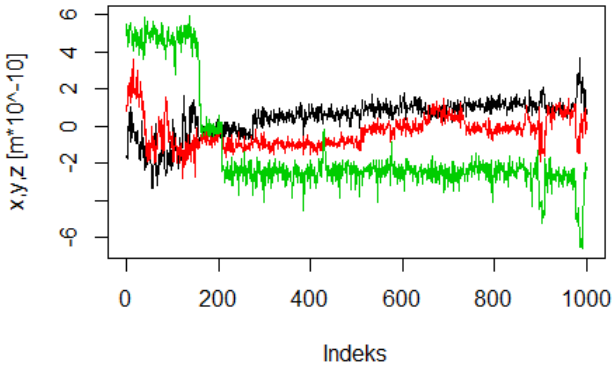
Først roteres og translateres billederne, så β-cyclodextrinet ligger i xy-planet og er centreret i punktet (0,0,0). Dette gøres ved først at translatere billede ved hjælp af centeret for β-cyclodextrinet, så dette er i origo. Herefter udregnes principal komponenterne for β-cyclodextrinet, hvorefter disse bruges til at rotere billedet.
Da principal komponenterne kun er entydige op til en ændring af fortegn, skal det sikres, at β-cyclodextrinet vender ens efter billederne er roteret i forhold til disse. Derfor foretages yderligere to transformationer af billedet. Først roteres, så atom nummer 126 har x-koordinat lig nul og positiv y-koordinat. Herefter spejles, så atom nummer 76 er først på x-aksen, hvis dette ikke allerede er tilfældet. R-kode for dette kan findes i appendiks A.1.

Når alle 1001 filer er blevet roteret og translateret i forhold til β-cyclodextrinet, ønskes det at anvende de resulterende billeder til at undersøge interaktionen mellem de to molekyler. For at gøre dette ses der på kolesterol-molekylets bevægelse i forhold til β-cyclodextrinet, ved både at undersøge molekylets forskydning langs de tre koordinatakser og molekylets vinkel til z-aksen.
Da det ikke vides, hvilken del af kolesterolet, der er vigtig i forhold til interaktionen med β-cyclodextrin, undersøges forskellige metoder til at beskrive kolesterol-molekylets bevægelse langs koordinatakserne.

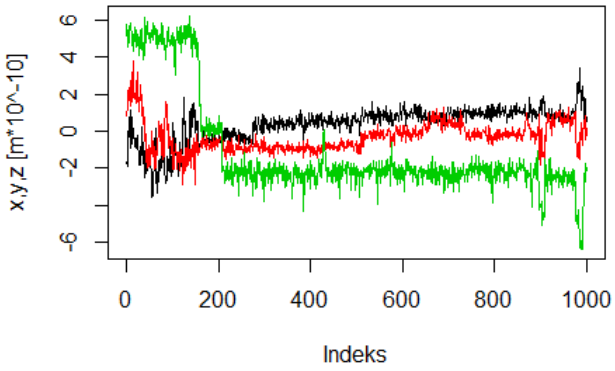
Først udregnes centret for kolesterol-molekylet, hvor atomerne vægtes på tre forskellige måder: et center udregnet med lige vægt på alle atomer, et massevægtet center samt et center, der kun er udregnet ud fra carbon-atomerne. Resultatet af disse kan ses på henholdsvis figur 2.2, 2.3 og 2.4. Hvordan disse er udregnet kan findes i appendiks A.3. Som forventet er de tre figurer meget ens.



Figur 2.2: Bevægelse af kolesterol langs de tre koordinatakser. Sort angiver x-aksen, mens rød angiver y-aksen og grøn angiver z-aksen.

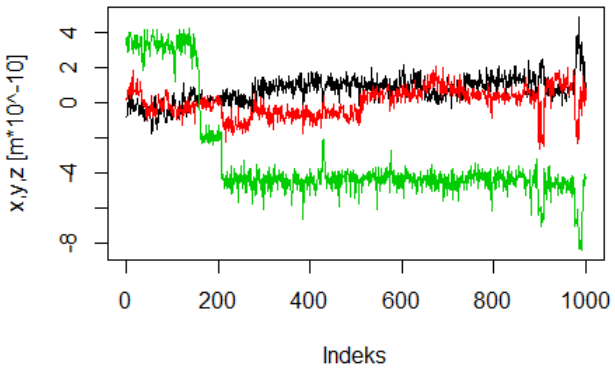


Figur 2.3: Bevægelse af massevægtet kolesterol langs de tre koordinatakser. Sort angiver x-aksen, mens rød angiver y-aksen og grøn angiver z-aksen.

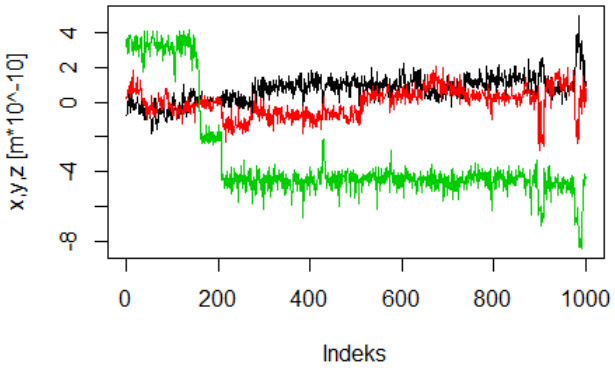


Figur 2.4: Bevægelse af carbon-atomerne i kolesterol langs de tre koordinatakser. Sort angiver x-aksen, mens rød angiver y-aksen og grøn angiver z-aksen.

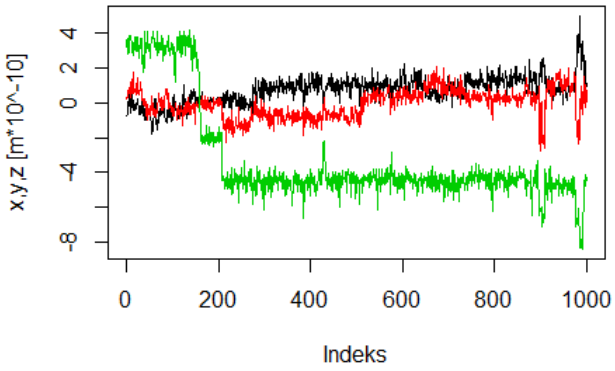
Kolesterol-molekylet kan deles op i to dele, en rigid ringstruktur samt en hale, hvilket kan ses på figur 2.9. Da der sker meget bevægelse i halen, som ikke nødvendigvis har med interaktionen mellem β-cyclodextrin og kolesterol at gøre, ønskes det at lave en beskrivelse af bevægelsen, som kun afhænger af bevægelsen i den rigide ringstruktur. Derfor udregnes et center for denne del på samme tre måder som ved centeret for hele kolesterol-molekylet. Disse kan ses på figur 2.5, 2.6 og 2.7. Udregning af disse kan findes i appendiks A.4. Igen ser de tre figurer, som forventet, meget ens ud. Bevægelsen langs x-aksen og y-aksen ser ikke ud til at have ændret sig i forhold til centrene for hele kolesterol-molekylet, men bevægelsen langs z-aksen er formindsket.



Figur 2.5: Bevægelse af den rigide del af kolesterol langs de tre koordinatakser. Sort angiver x-aksen, mens rød angiver y-aksen og grøn angiver z-aksen.

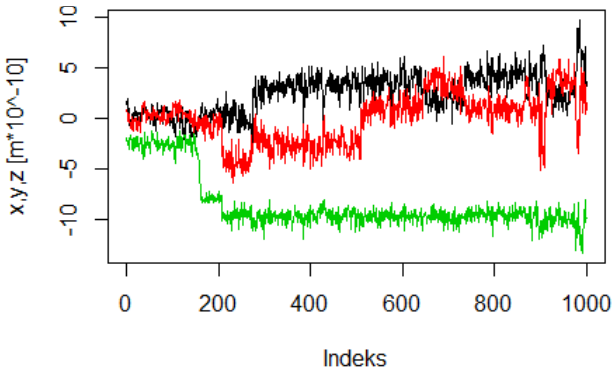


Figur 2.6: Bevægelse af den massevægtede rigide del af kolesterol langs de tre koordinatakser. Sort angiver x-aksen, mens rød angiver y-aksen og grøn angiver z-aksen.



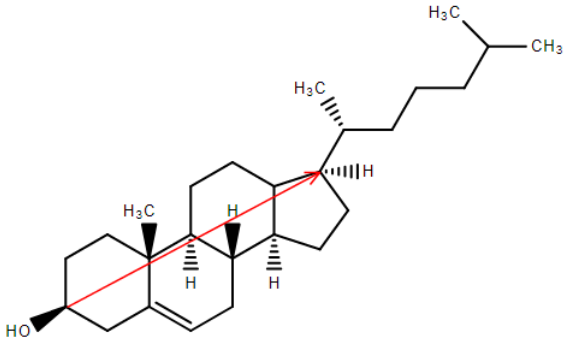
Figur 2.7: Bevægelse af carbonatomerne i den rigide del af kolesterol langs de tre koordinatakser. Sort angiver x-aksen, mens rød angiver y-aksen og grøn angiver z-aksen.

Herefter undersøges bevægelsen af oxygen-atomet i kolesterol-molekylet, da dette har en yderlig position. Resultatet af denne kan ses på figur 2.8. Udregningen for oxygen-atomets bevægelse kan findes i appendiks A.5. Bevægelsen af denne er noget større end bevægelsen af centrene for hele kolesterol-molekylet og for den rigide ringstruktur i kolesterol-molekylet, hvilket ikke er overraskende grundet den yderlige placering af oxygen-atomet i strukturen af kolesterol-molekylet.

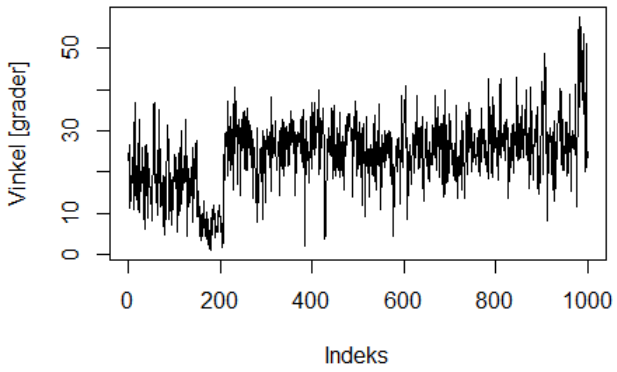


Figur 2.8: Bevægelse af iltatomet i kolesterolmolekylet langs de tre koordinatakser. Sort angiver x-aksen, mens rød angiver y-aksen og grøn angiver z-aksen.

Til sidst ønskes det at undersøge, hvordan vinklen mellem en vektor, som går gennem den rigide del af kolesterol-molekylet, og z-aksen ændrer sig. Vektoren, som anvendes til dette, kan ses på figur 2.9. Variationen af denne vinkel kan ses på figur 2.10. Hvordan vinklen er udregnet kan findes i appendiks A.6.



Figur 2.9: Vektoren, der anvendes til at beskrive den rigide del af kolesterol-molekylet.

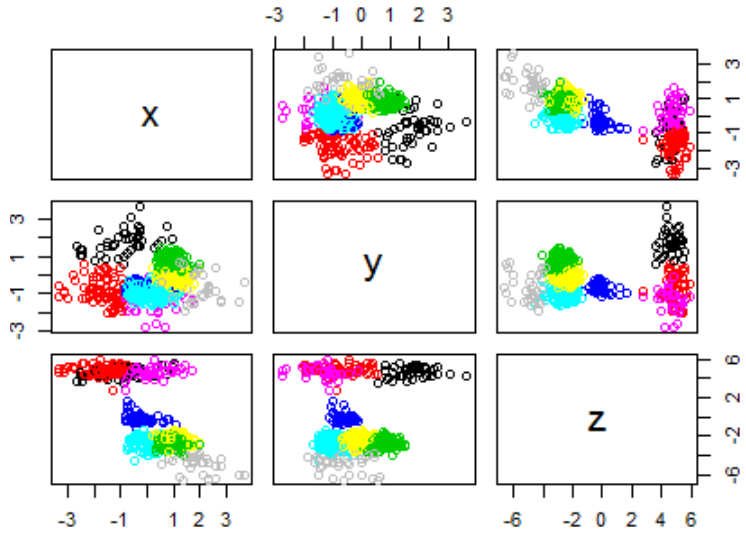


Figur 2.10: Vinklen mellem vektor gennem kolesterol-molekylet og z-aksen.

2.3 Klyngeanalyse

2.3.1 K-middel klyngedannelse

Til K-middel klyngedannelse anvendes funktionen `kmeans()` på de otte forskellige metoder til beskrivelse af kolesterol-molekylets bevægelse. Som eksempel ses på figur 2.11 resultatet af K-middel klyngedannelse af det massevægtede center for hele kolesterol-molekylet. Antallet af klynger, $K = 8$, er valgt for visuelt bedre at kunne adskille de forskellige klynger på plottet. For at opnå den klyngedannelse med mindst *within-cluster point scatter*, $W(C)$, itereres klyngedannelsen med ti forskellige initielle klyngecentre, som vælges tilfældigt af funktionen. Dette giver $W(C) = 531,7$.



Figur 2.11: Klynger opnået ved K-middel klyngedannelse af det massevægtede center af kolesterolmolekylet og $K = 8$.

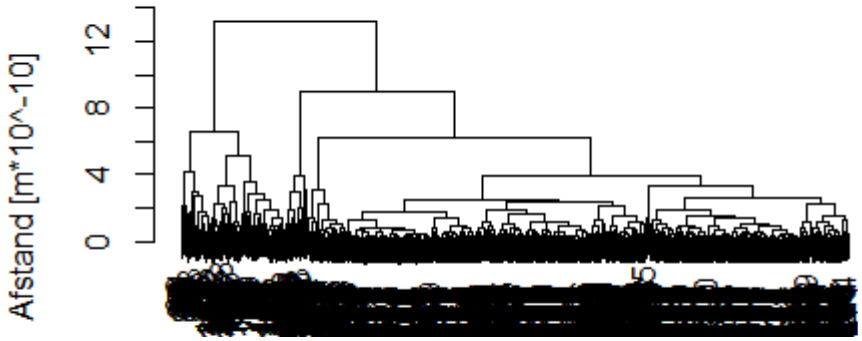
Det ønskes nu at anvende K-middel klyngedannelse til at reducere de 1001 filer til et mindre antal. Derfor anvendes funktionen, som vist i eksemplet ovenfor, på de otte forskellige beskrivelser, men i stedet anvendes $K = 10$. De resulterende klyngecentre anvendes da til at finde et billede, som beskriver hele klyngen. Dette billede vælges til at være det, som ligger tættest på det specifikke klyngecenter, hvilket giver reduceringer af datasættet som ses i tabel 2.1. Plots af de resulterende klynger fra de forskellige klyngedannelser kan findes i appendiks B.1.

Beskrivelse	**Billeder**	$W(C)$
Center	17, 37, 76, 122, 173, 267, 388, 895, 951, 980	451,58
Center, mv	17, 37, 76, 122, 173, 267, 287, 554, 647, 980	446,39
Center, c	17, 88, 97, 122, 173, 308, 447, 815, 947, 980	430,66
Center, r	16, 63, 191, 213, 348, 465, 668, 750, 807, 981	431,99
Center, r, mv	16, 63, 191, 213, 302, 465, 630, 642, 702, 981	438,2
Center, r, c	16, 63, 191, 213, 399, 465, 630, 644, 702, 981	440,08
Oxygen	49, 192, 239, 352, 356, 529, 717, 850, 968, 983	1539,34
Vinkel	4, 79, 113, 381, 472, 542, 603, 873, 914, 985	1550,75

Tabel 2.1: Reducering af datasæt ved brug af K-middel klyngedannelse. Forkortelserne r, c og mv står for henholdsvis rigid, carbon og massevægtet.

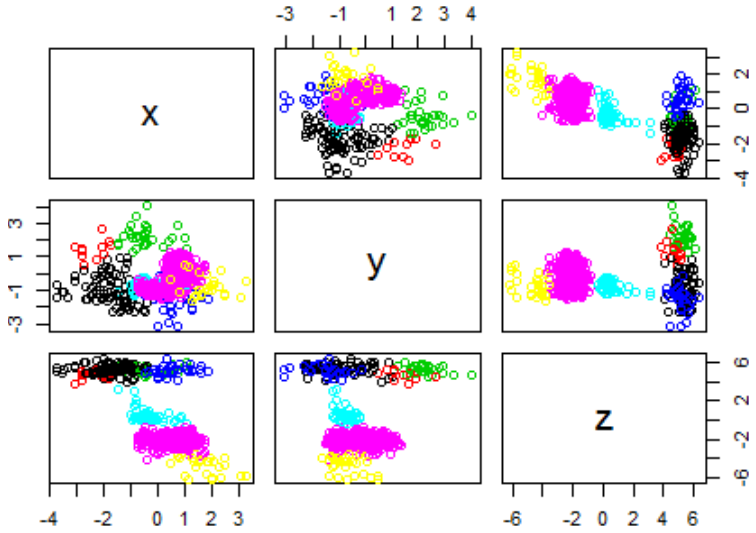
2.3.2 Hierarkisk klyngedannelse

Til hierarkisk klyngedannelse anvendes funktionen `hclust()`, som benytter sig af agglomerativ klyngedannelse, på de otte forskellige metoder til beskrivelse af kolesterol-molekylets bevægelse. Som eksempel ses på figur 2.12 det resulterende dendrogram fra en hierarkisk klyngedannelse af det massevægtede center for hele kolesterol-molekylet.



Figur 2.12: Dendrogram fra agglomerativ klyngedannelse af det massevægtede center af kolesterol-molekylet.

Ved at lave et snit ved en afstand på cirka 4 m·10^{-10} opnås syv klynger, som kan ses på figur 2.13.



Figur 2.13: Klynger opnået ved agglomerativ klyngedannelse af det massevægtede center af kolesterol-molekylet.

Den hierarkiske klyngedannelse anvendes nu til at reducere de 1001 filer til et mindre antal ud fra de otte forskellige beskrivelser, ved samme metode som gen-

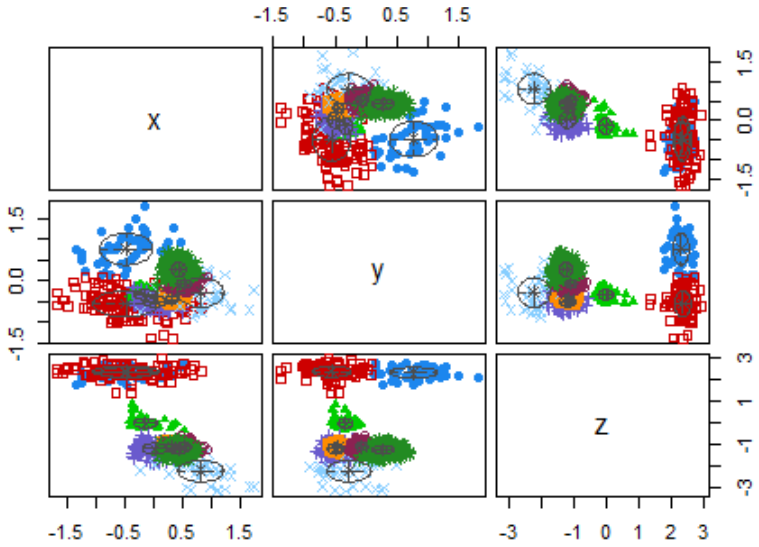
nemgået i eksemplet ovenfor. Klyngecentrene fra den hierarkiske klyngedannelse anvendes til at finde de billeder, som ligger tættest på disse, hvilket giver resultaterne i tabel 2.2. Plots af både dendrogrammer og de resulterende klynger kan findes i appendiks B.2.

Beskrivelse	**Billeder**	**Antal**
Center	36, 79, 85, 117, 159, 557, 576, 990	8
Center, mv	36, 85, 105, 153, 472, 553, 576, 901, 992	9
Center, c	15, 88, 98, 105, 553, 576, 990,	7
Center, r	105, 351, 431, 754, 977, 983, 986, 992	8
Center, r, mv	105, 351, 431, 506, 754, 901, 983, 986, 992	9
Center, r, c	79, 160, 431, 754, 832, 902, 977, 983, 991, 992	10
Oxygen	158, 171, 228, 265, 360, 632, 659, 803, 976, 983, 987, 994	12
Vinkel	52, 60, 174, 283, 374, 774, 858, 867, 979, 990, 998	11

Tabel 2.2: Reducering af datasæt ved brug af hierarkisk klyngedannelse. Forkortelserne r, c og mv står for henholdsvis rigid, carbon og massevægtet.

2.3.3 Mixture modeling

Ved *mixture modeling* anvendes funktionen `Mclust()` på de otte forskellige metoder til beskrivelse af kolesterol-molekylets bevægelse. Derudover anvendes funktionen på beskrivelsen af centeret, som i afsnit 2.3.1 gav den mindste *within cluster point scatter*, sammen med vinklen mellem kolesterol-molekylet og z-aksen. Som eksempel ses på figur 2.14 resultatet af *mixture modeling* af det massevægtede center for hele kolesterol-molekylet. Der opnås en VVI-model med otte klynger.



Figur 2.14: Klynger opnået ved *mixture modeling* af det massevægtede center af kolesterol-molekylet.

Resultaterne af *mixture modeling* anvendes nu til at reducere de 1001 filer til et mindre antal på de otte oprindelige beskrivelser og undersøgelse af centeret af carbon-atomerne i kolesterol-molekylet og vinklen. Klyngecentrene fra *mixture modeling* anvendes til at finde de billeder, som ligger tættest på disse, hvilket giver resultaterne i tabel 2.3. Plots af de resulterende klynger kan findes i appendiks B.3.

Beskrivelse	Billeder	Model	Antal	BIC
Center	17, 52, 208, 276, 554, 875, 980	VVI	7	-5984,7
Center, mv	27, 117, 173, 324, 421, 647, 735, 815, 980	VVI	9	-6157,9
Center, c	19, 52, 173, 223, 421, 584, 731, 859, 980	VEV	9	-5858,8
Center, r	74, 191, 263, 299, 341, 630, 668, 737, 993	VEI	9	-6360,7
Center, r, mv	74, 191, 263, 299, 341, 614, 630, 659, 993	VEI	9	-6763,9
Center, r, c	74, 191, 263, 299, 341, 614, 630, 659, 993	VEI	9	-6437,6
Oxygen	49, 191, 213, 349, 488, 590, 716	VVV	7	-10434,3
Vinkel	201, 510, 984	E	3	-7070,1
Center+vinkel	80, 173, 223, 465, 554, 647, 735, 905	VVV	8	-12003

Tabel 2.3: Reducering af datasæt ved brug af *mixture modeling*. Forkortelserne r, c og mv står for henholdsvis rigid, carbon og massevægtet.

Diskussion

Fra de resulterende figurer fra K-middel klyngedannelse, som kan findes i appendiks B.1, er det tydeligt, at de beskrivelser af bevægelsen af kolesterol-molekylet, som ligner hinanden, har meget lignende resulterende klyngedannelser. Dette var muligvis også forventeligt, eftersom at tidsrækkeplots fra den indledende dataanalyse også viste store ligheder.
I reduktionen er det dog ikke præcist de samme billeder, der er blevet valgt. Dette skyldes, at der i klyngedannelsen ikke skal meget til for at ændre placeringen af klyngecenteret, hvilket kan resultere i, at et andet billede bedre beskriver klyngen.

Ved brugen af hierarkisk klyngedannelse var det ønsket, at de visuelle billeder af klyngedannelsen skulle give et mere sigende indblik i, hvor mange klynger der skule vælges. Det har vist sig, at brugen af hierarkisk klyngedannelse ikke har haft den tilsigtede effekt. Dette kommer sig af, at der ikke forefindes et naturligt hierarki i datasættet, hvilket også er tydeligt på dendrogrammerne, da der ikke er et åbentlyst sted, hvor det giver mening at lave snittet.
Det er dog en mulighed, at den hierarkiske klyngedannelse havde givet et mere sigende udfald, hvis man, ud fra et kemisk synspunkt, kunne sige noget om, hvor snittet skulle lægges. Algoritmen opdeler stadig klyngerne efter hensigten, men da man ikke definitivt kan sige noget fornuftigt om, hvor snittet skal lægges, så opnås der ikke yderligere fordele fremfor brugen af K-middel klyngedannelse.
Der er igennem projektet brugt den samme type linkage, for at gøre sammenligning af resultater lettere. Hvis der fandtes forskning, der kunne understøtte valget af linkage, ville det være muligt at tage et mere optimalt valg i forhold til datasættet.

I forhold til den hierarkiske klyngedannelse giver *mixture modeling* et mere matematisk indblik i valget af klyngecentre, eftersom at disse vælges på baggrund af modellernes BIC. Resultaterne fra *mixture modeling* giver klyngedannelserne for de tre beskrivelser af centeret for den rigide del af kolesterol-molekylet tre nærmest helt ens resultater. Det samme gør sig gældende for de resulterende reduktioner. Derimod giver klyngedannelserne af de tre beskrivelser af centeret for kolesterol

tre forskellige modeller, og reduktionerne har også kun få sammenfald.

Det kan også overvejes, hvorvidt de beskrivelser, der er lavet af kolesterol-molekylet giver en god beskrivelse af, hvad der i virkeligheden sker i forhold til interaktionen mellem β-cyclodextrin og kolesterol. Da denne interaktion danner grundlag for igangværende forskning, er meget stadig uvist om denne. Derfor kunne nye beskrivelser af molekylet give andre resultater, der også kunne bidrage til en forståelse af denne interaktion.

Som videre arbejde kan det også overvejes, om der skal laves ændringer i den måde simulationen af molekylerne er udført på. Der kunne eksempelvis undersøges, om det giver mening, at der er vand omkring molekylet, eller om det giver en effekt, at der laves kemiske modificeringer på β-cyclodextrinet.

Konklusion

Interaktionen mellem kolesterol og β-cyclodextrin har i nyere tid været et område for meget forskning. Dette grunder sig i en formodning om at β-cyclodextrin har en positiv virkning på kolesterol-baserede sygdomme.

I dette projekt undersøges computersimuleret data, som er genereret for at sige noget specifik om den interaktion, der er mellem β-cyclodextrin og kolesterol. I et forsøg på at finde yderligere information om interaktionen undersøges dette datasæt for at finde underliggende strukturer. Dette er gjort med henblik på at kunne reducere de ellers meget store datasæt, så kompleksiteten af det videre arbejde eventuelt kan nedsættes.

Der er inden for emnet *unsupervised learning* blevet anvendt forskellige metoder til at analysere de ovennævnte underliggende strukturer. De følgende metoder er blevet brugt: K-middel klyngedannelse, hierarkisk klyngedannelse og *mixture modeling*. Igennem databehandling kan det konstateres, at datasættes underliggende strukturer kan beskrives ved hjælp af de anvendte metoder, som videre kan benyttes til at finde reduceringer af datasættet. Det formodes, at disse reduceringer giver et bedre indblik i datasættet, end hvis en tilfældig reducering blev valgt, men dette er ikke blevet bekræftet.

Bibliografi

[1] Robert P. Erickson og Maria Teresa Fiorenza. "A hopeful therapy for Niemann-Pick C diseases". I: *The Lancet* 390 (2017), s. 1720–1721.

[2] Chris Fraley m.fl. *mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation*. Tek. rap. 59. Department of Statistics, University of Washington, 2012, s. 57.

[3] Trevor Hastie, Robert Tibshirani og Jerome Freidman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2. ed. Springer-Verlag New York, 2009.

[4] Gareth James m.fl. *An Introduction to Statistical Learning: with Applications in R*. 1. ed. Springer-Verlag New York, 2013.

Appendiks A

R-kode

R-koden og datasættet brugt i dette projekt kan findes på følgende link:
https://github.com/amsoba/bcd_chol

A.1 onePicture

```
onePicture <- function(file,rotxy=TRUE){
    require("bio3d")
    ch <- read.pdb2(file)$atom
    ch <- ch[,c("eleno","elety","resid","resno","x","y","z","elesy")]
    for(i in c(2,3,8)) ch[,i] <- factor(ch[,i])
    GAxyz <- ch[ch$resid=="4GA",c("x","y","z")]
    p1 <- princomp(GAxyz)
    rotation <- p1$loadings[,]
    center <- apply(GAxyz,2,mean)
    ch[,c("x","y","z")] <- (as.matrix(ch[,c("x","y","z")])-center)%*%rotation
    if(ch[42,7]>0) ch[,7] <- -ch[,7]
    if(rotxy){
        ref <- as.numeric(ch[126,5:6])
        ref <- ref/sqrt(sum(ref^2))
        rot <- matrix(c(ref[2],-ref[1],ref[1],ref[2]),2)
        ch[,5:6] <- cbind(ch[,5],ch[,6])%*%rot
        if(ch[76,5]>0) ch[,5] <- -ch[,5]
    }
    ch
}
```

I nedenstående er outputtet fra henholdsvis funktionen `head()` og funktionen `tail()` fra en tilfældig fil nedskrevet, efter brug af funktionen `onePicture()`. Dette er for at give et billede af, hvad de forskellige variable i filerne dækker over. De forskellige variable dækker over følgende:

- `eleno`: Angiver elementets nummer.
- `elety`: Angiver en mere specifik beskrivelse af atomets placering i molekylet.
- `resid`: Angiver hvilket molekyle elementet tilhører.
- `resno`: Angiver om elementet tilhører en af de syv sukkerringe i β-cyclodextrin eller om det tilhører kolesterol.
- `x,y,z`: Angiver koordinaterne for elementets placering.
- `elesy`: Angiver elementets type.

I de følgende tabeller ses eksempler på hvilke værdier de forskellige variable antager.

	eleno	elety	resid	resno	x	y	z	elesy
1	1	C1	4GA	1	5.148584	-3.1891730	-0.4982520	C
2	2	C2	4GA	1	5.995125	-2.6310152	0.6376830	C
3	3	C3	4GA	1	5.745428	-1.1486243	0.9273920	C
4	4	C4	4GA	1	5.708309	-0.2408389	-0.3237437	C
5	5	C5	4GA	1	4.790478	-0.9852701	-1.3855117	C
6	6	C6	4GA	1	4.733589	-0.1587366	-2.7450042	C

Tabel A.1: Resultat fra `head()`.

	eleno	elety	resid	resno	x	y	z	elesy
216	216	H41	UNK	900	0.75784251	3.232852	-6.9011710	H
217	217	H42	UNK	900	1.37219351	3.756179	-5.2850967	H
218	218	H43	UNK	900	-0.04469139	2.786293	-5.4882956	H
219	219	H44	UNK	900	-1.90272008	1.250973	-1.9821232	H
220	220	H45	UNK	900	-0.79677143	2.309781	-2.7680721	H
221	221	H46	UNK	900	-1.16582176	2.425850	-0.9929772	H

Tabel A.2: Resultat fra `tail()`.

A.2 morePictures

```
files <- list.files()
OK <- c()
for(i in 1:length(files)){
    x <- files[i]
    n <- nchar(x)
    if(substr(files[i],n-3,n)==".pdb") OK <- c(OK,i)
}
files <- files[OK] # vælger filer af typen .pdb
fileno <- function(x){
    n <- nchar(x)
    substr(x,10,n-4)
}
no <- as.numeric(sapply(files,fileno))
files <- files[order(no)] # filer ordnes efter nummer
pictures <- list()
for(x in files) pictures <- c(pictures,list(onePicture(x)))
```

A.3 Centre for kolesterol

Center med lige vægt på alle atomer:

```
centerh <- c()
for(x in pictures){
  xyz <- x[x$resno==900,5:7]
  centerh <- rbind(centerh,apply(xyz,2,mean))
}
```

Massevægtet center:

```
centerm <- c()
for(x in pictures){
  c <- x[x$resno==900&x$elesy=="C",5:7]
  o <- (8/6)*x[x$resno==900&x$elesy=="O",5:7]
  h <- (1/12)*x[x$resno==900&x$elesy=="H",5:7]
  vaegt <- dim(c)[1]+8/6+dim(h)[1]/12
  centerm <- rbind(centerm,apply(rbind(c,h,o),2,sum)/vaegt)
}
```

Center udregnet ud fra carbon-atomerne:

```
centerc <- c()
for(x in pictures){
  xyz <- x[x$resno==900&x$elesy=="C",5:7]
  centerc <- rbind(centerc,apply(xyz,2,mean))
}
```

A.4 Centre for rigid del af kolesterol

Center for den rigide del af kolesterol med lige vægt på alle atomer:

```
rigidh <- c()
for (x in pictures){
  xyz <- x[c(156:172,174:175,193:214,216:221), 5:7]
  rigidh <- rbind(rigidh, apply(xyz,2,mean))
}
```

Massevægtet center for den rigide del af kolesterol:

```
rigidm <- c()
for(x in pictures){
  c <- x[c(156:172,174:175),5:7]
  h <- (1/12)*x[c(193:214,216:221),5:7]
  vaegt <- dim(c)[1]+dim(h)[1]/12
  rigidm <- rbind(rigidm,apply(rbind(c,h),2,sum)/vaegt)
}
```

Center udregnet ud fra carbon-atomerne i den rigide del af kolesterol:

```
rigidc <- c()
for (x in pictures){
  xyz <- x[c(156:172,174:175),5:7]
  rigidc <- rbind(rigidc, apply(xyz,2,mean))
}
```

A.5 Oxygen

```
oxygen <- c()
for(x in pictures){
  xyz <- x[173,5:7]
  oxygen <- rbind(oxygen,xyz)
}
```

A.6 Vinkel mellem vektor i kolesterol og z-aksen

```
angle <- c()
for(x in pictures){
  a <- x[x$eleno==171,5:7]
  b <- x[x$eleno==156,5:7]
  vec <- c(b[,1]-a[,1],b[,2]-a[,2],b[,3]-a[,3])
  z <- c(0,0,1)
  theta <- (180/pi)*acos((vec%*%z)/(sqrt(vec[1]^2+vec[2]^2+vec[3]^2)
            *sqrt(z[1]^2+z[2]^2+z[3]^2)))
  angle <- rbind(angle,theta)
}
```

A.7 Reduceringer

A.7.1 K-middel klyngedannelse

Først udføres K-middel klyngedannelse på det ønskede data.

```
clustxx = kmeans(x, 10, nstart=10)
```

Herefter findes de punktet som er tættest på klyngecentrene.

```
eud <- function(a,b){
  (a[1]-b[1])^2+(a[2]-b[2])^2+(a[3]-b[3])^2
}

mindist <- function(x,c){
    di <- 100
    for (i in 1:1001){
        d <- eud(x[i,],c)
        if (d>di) next
        di <- d
    }
    di
}

kmidx <- c()
for (i in 1:20){
  kmidx <- rbind(kmidx,mindist(x,clustx$centers[i,]))
}
```

A.7.2 Hierarkisk klyngedannelse

Først udføres den hierarkiske klyngedannelse på det ønskede data. `K` er her det valgte antal klynger ud fra dendrogrammet.

```
hclustx = hclust(dist(x), method = "complete")
cutx = cutree(hclustx, K)
```

Herefter udregnes centrene for klyngerne.

```
hcenter <- function(data){
    hcentre <- c()
    for (i in 1:max(data)){
        k <- c()
       for (j in 1:1001){
            if (data[j,4]==i){
            ny <- data[j,1:3]
            k <- rbind(k,ny)
            k <- apply(k,2,mean)
            }
        }
        hcentre <- rbind(hcentre, k)
    }
hcentre
}
```

Til sidst findes de punkter, som er tættest på klyngecentrene.

```
    hx <- cbind(x, as.data.frame(cutx))
hierx <- c()
for (i in 1:max(cutx)){
  hierx <- rbind(hierx,mindist(x,hcenter(hx)[i,]))
}
```

A.7.3 Mixture modeling

Først udføres mixture modeling på det ønskede data. Her tilføjes alle de mulige modeller til `modelNames`, for at funktionen fitter alle modeller, for herefter at vælge den med den største BIC.

```
modx = Mclust(x, modelNames = c("EII","VII","EEI","VEI","EVI",
  "VVI","EEE","EEV","VEV","VVV"))
```

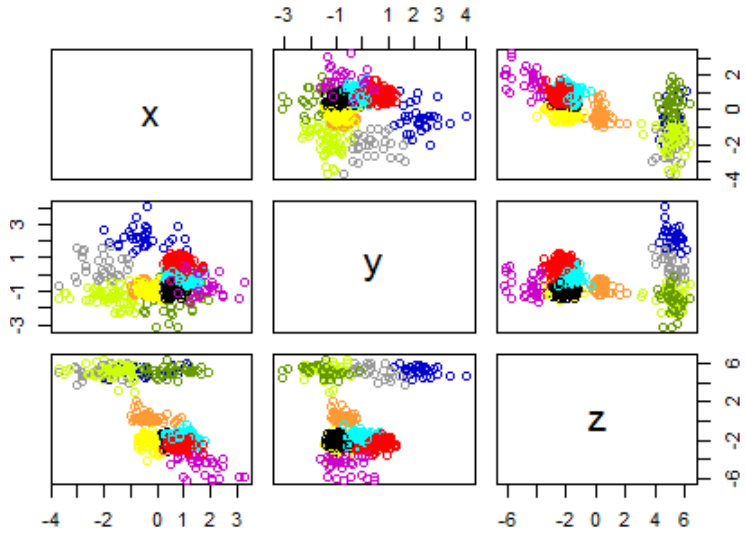
Derefter benyttes nedenstående til at finde de punkter, som er tættest på de opnåede klyngecentre.

```
antalklynger <- modx$G
hvor <- modx$classification
indekser <- 1:length(hvor)
means <- modx$par$mean
vars <- modx$par$variance$sigma
klyngecentre <- c()
for(i in 1:antalklynger){
    klyngei <- indekser[hvor==i]
    meani <- means[,i]
    variInvers <- solve(vars[,,i])
    disti <- Inf
    kmin <- 0
    for(k in klyngei){
        h <- matrix(x[k,]-meani,ncol=1)
        d <- t(h)%*%variInvers%*%h
        if(d>disti) next
        kmin <- k
        disti <- d
    }
klyngecentre <- c(klyngecentre,kmin)
}
```

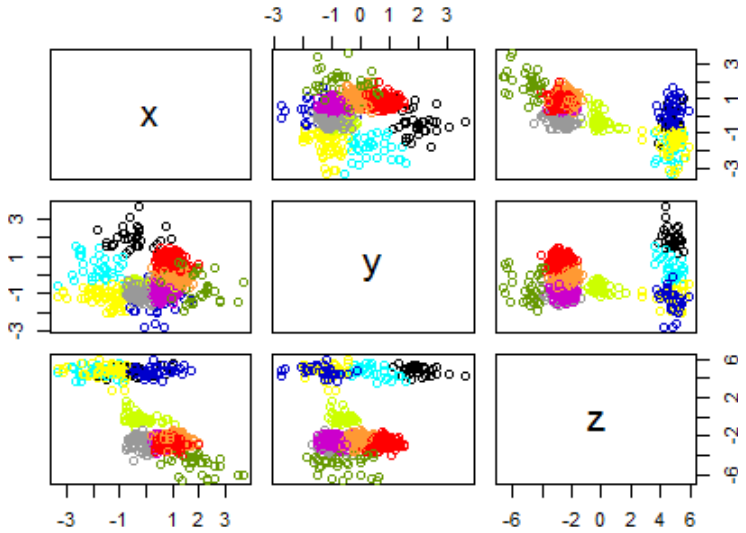
Appendiks B

Figurer

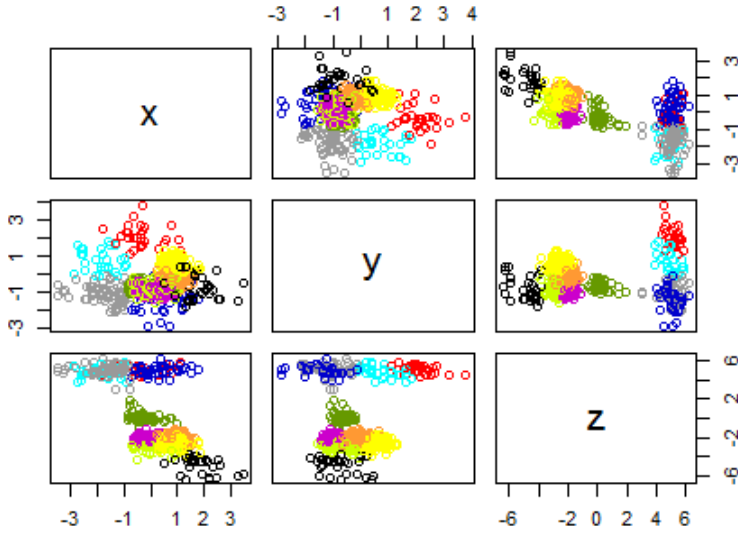
B.1 K-middel klyngedannelse



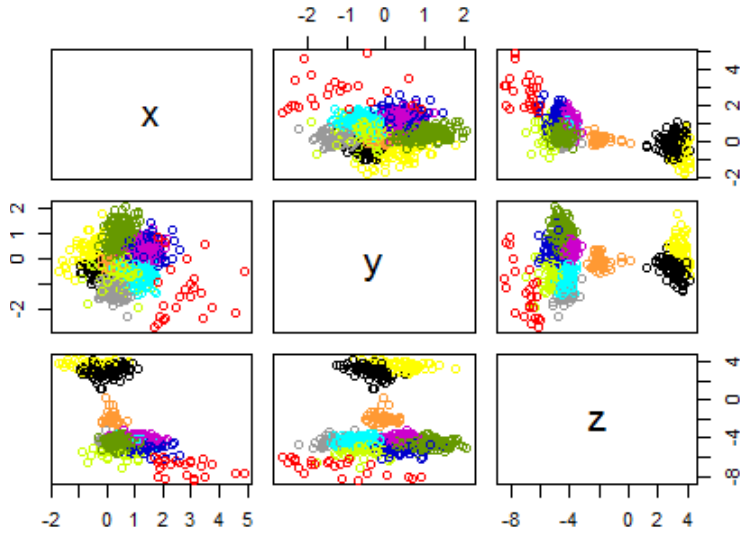
Figur B.1: Klynger opnået ved K-middel klyngedannelse af centeret af kolesterol-molekylet.



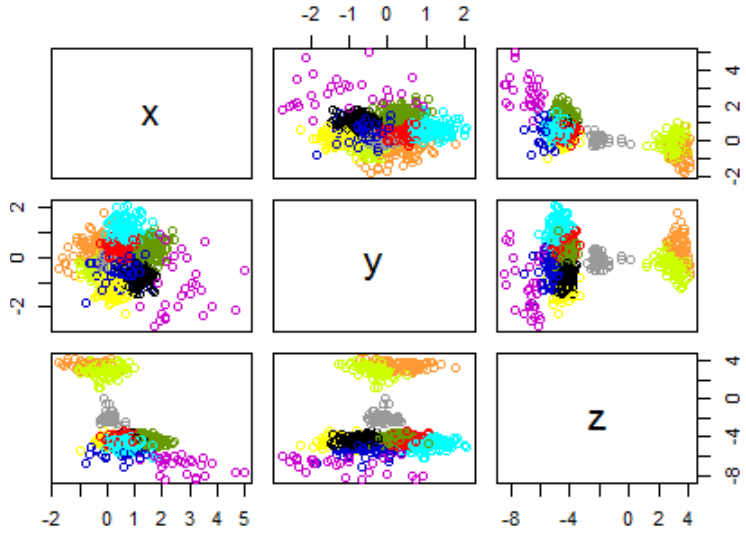
Figur B.2: Klynger opnået ved K-middel klyngedannelse af det massevægtede center af kolesterolmolekylet.



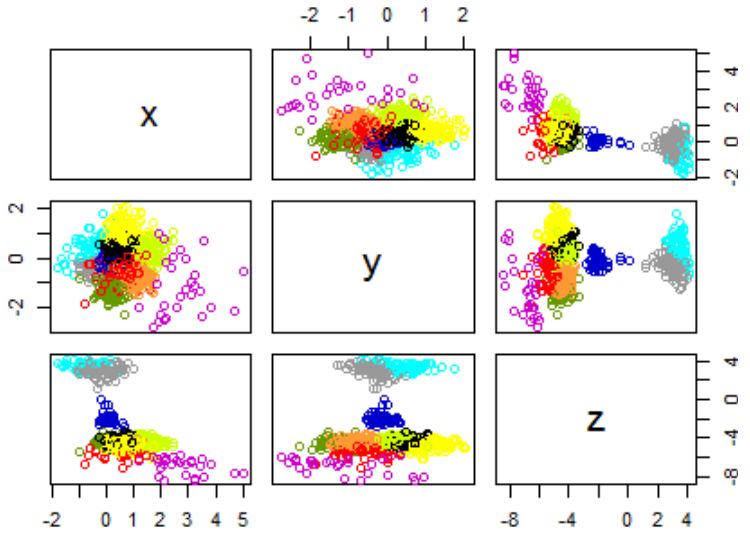
Figur B.3: Klynger opnået ved K-middel klyngedannelse af centeret af carbon-atomerne i kolesterolmolekylet.



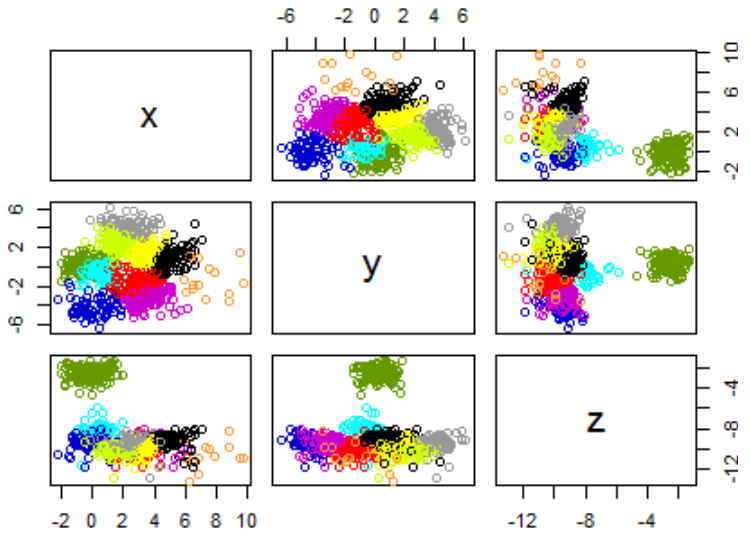
Figur B.4: Klynger opnået ved K-middel klyngedannelse af centeret af den rigide del af kolesterol-molekylet.



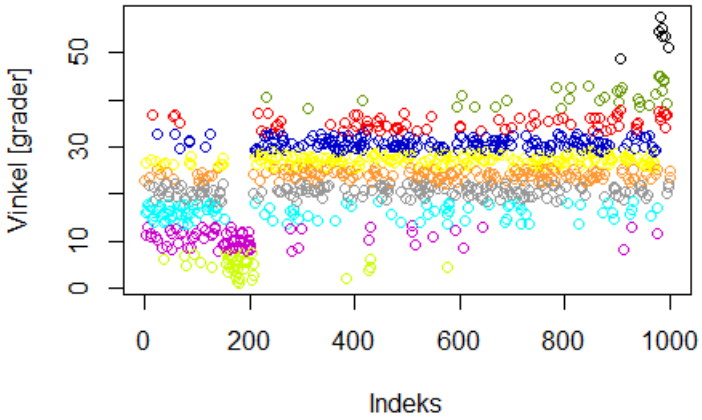
Figur B.5: Klynger opnået ved K-middel klyngedannelse af det massevægtede center af den rigide del af kolesterol-molekylet.



Figur B.6: Klynger opnået ved K-middel klyngedannelse af centeret af carbon-atomerne i den rigide del af kolesterol-molekylet og $K = 8$.



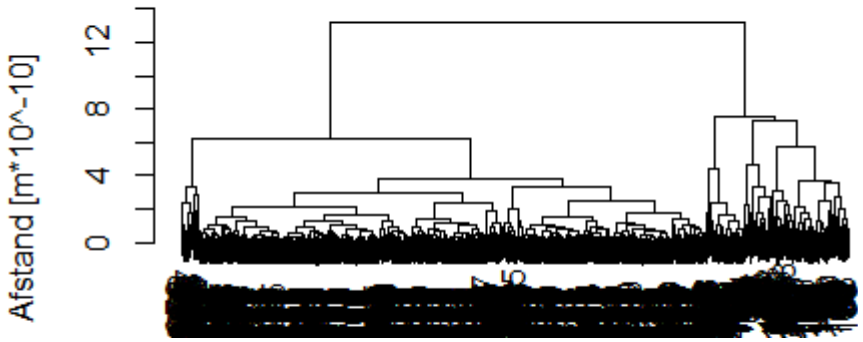
Figur B.7: Klynger opnået ved K-middel klyngedannelse af oxygen-atomet i kolesterol-molekylet og $K = 8$.



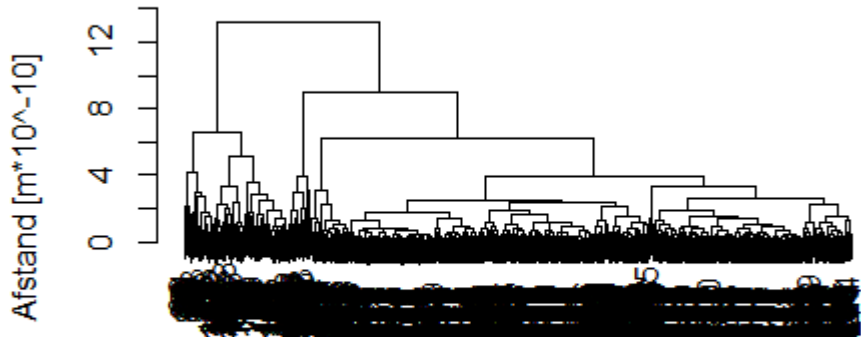
Figur B.8: Klynger opnået ved K-middel klyngedannelse af vinklen mellem vektoren i kolesterol-molekylet og z-aksen.

B.2 Hierarkisk klyngedannelse

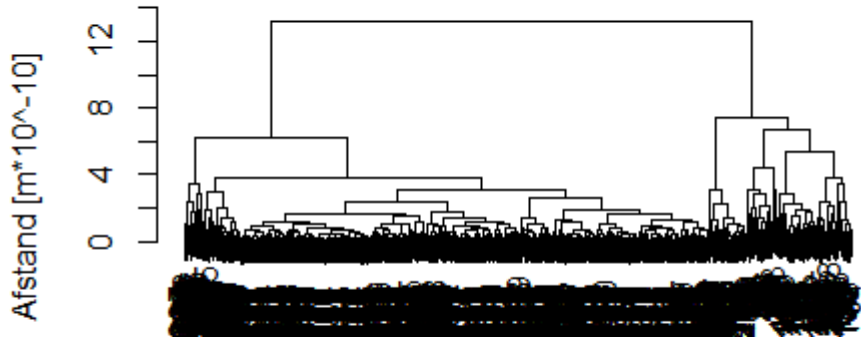
B.2.1 Dendrogrammer



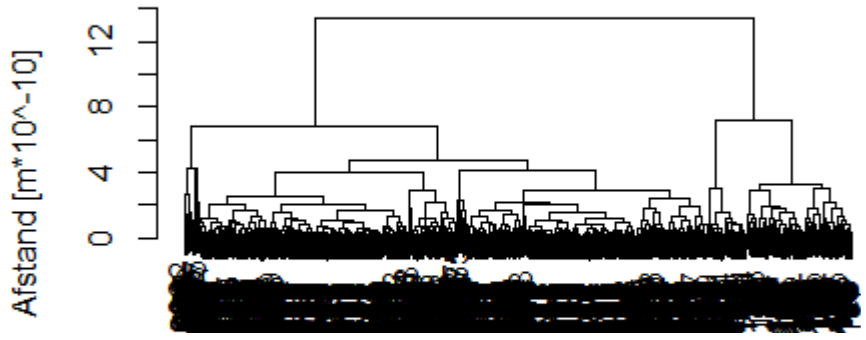
Figur B.9: Dendrogram fra agglomerativ klyngedannelse af centeret af kolesterol-molekylet.



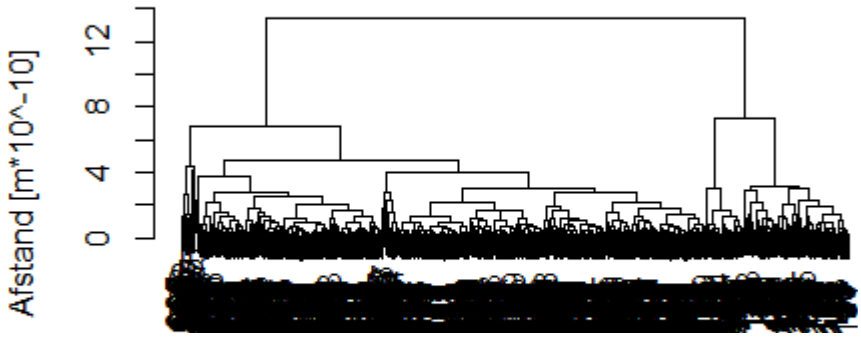
Figur B.10: Dendrogram fra agglomerativ klyngedannelse af det massevægtede center af kolesterol-molekylet.



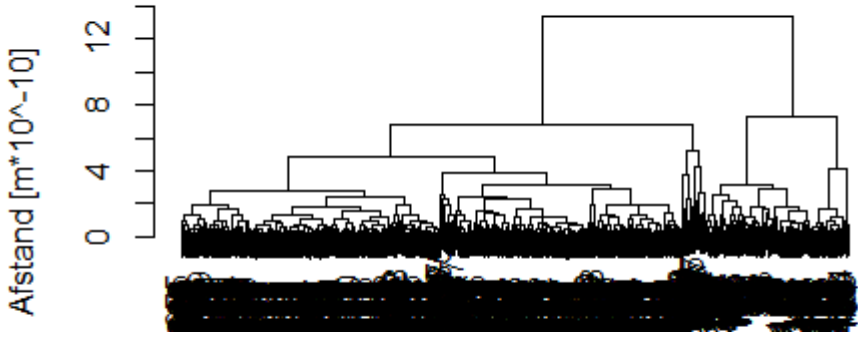
Figur B.11: Dendrogram fra agglomerativ klyngedannelse af centeret af carbon-atomerne af kolesterol-molekylet.



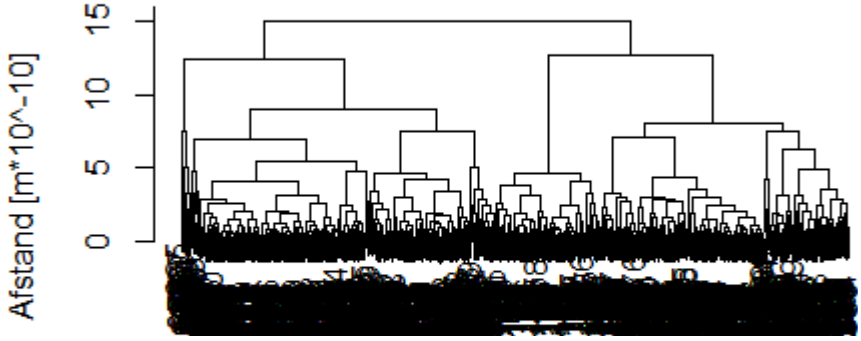
Figur B.12: Dendrogram fra agglomerativ klyngedannelse af centeret af den rigide del af kolesterol-molekylet.



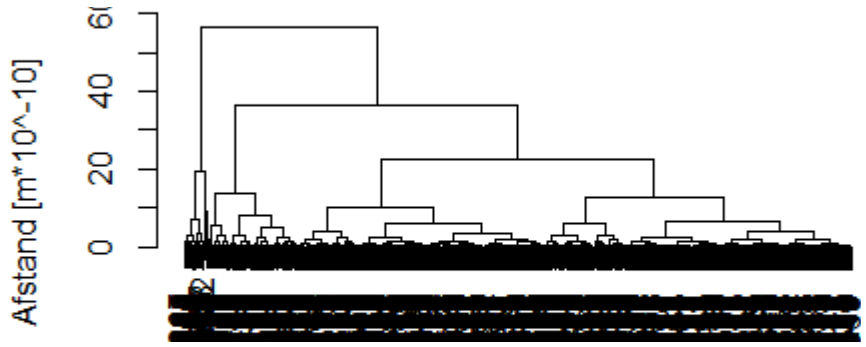
Figur B.13: Dendrogram fra agglomerativ klyngedannelse af det massevægtede center af den rigide del af kolesterol-molekylet.



Figur B.14: Dendrogram fra agglomerativ klyngedannelse af centeret af carbon-atomerne i den rigide del af kolesterol-molekylet.

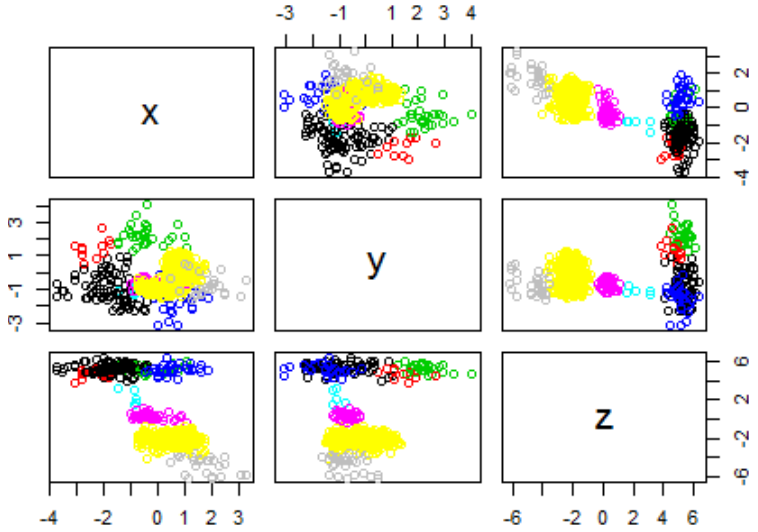


Figur B.15: Dendrogram fra agglomerativ klyngedannelse af oxygen-atomet i kolesterol-molekylet.

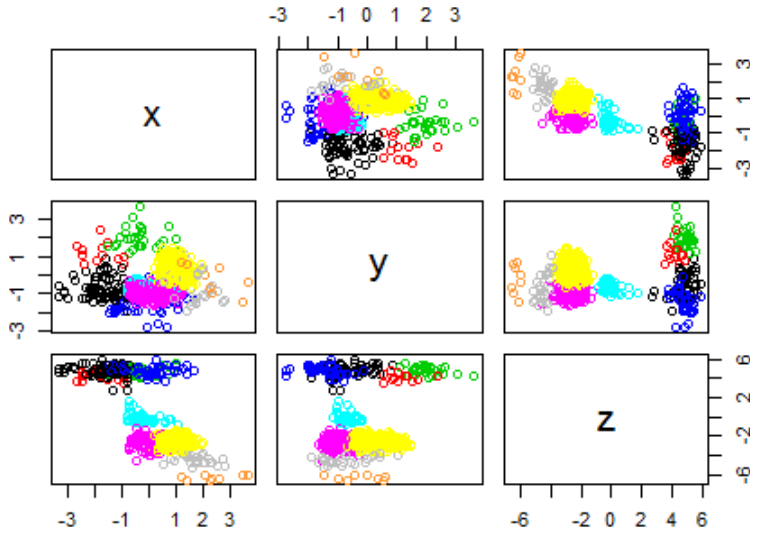


Figur B.16: Dendrogram fra agglomerativ klyngedannelse af vinklen mellem vektoren i kolesterol-molekylet og z-aksen.

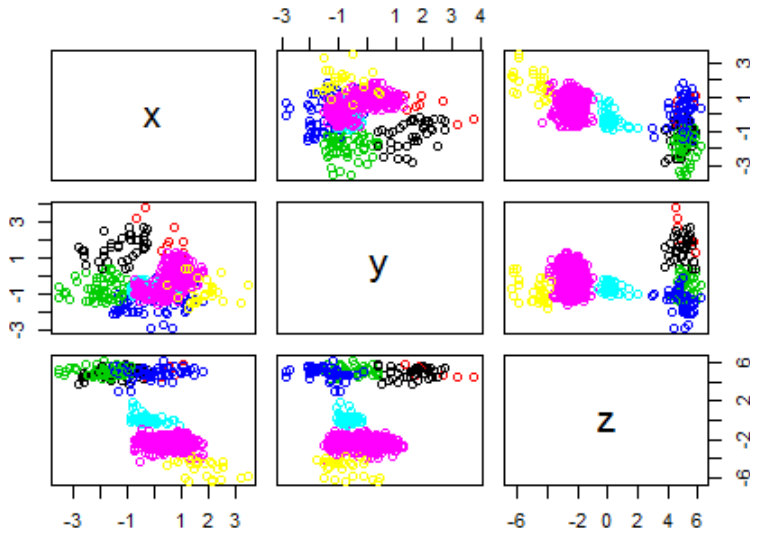
B.2.2 Klyngedannelse



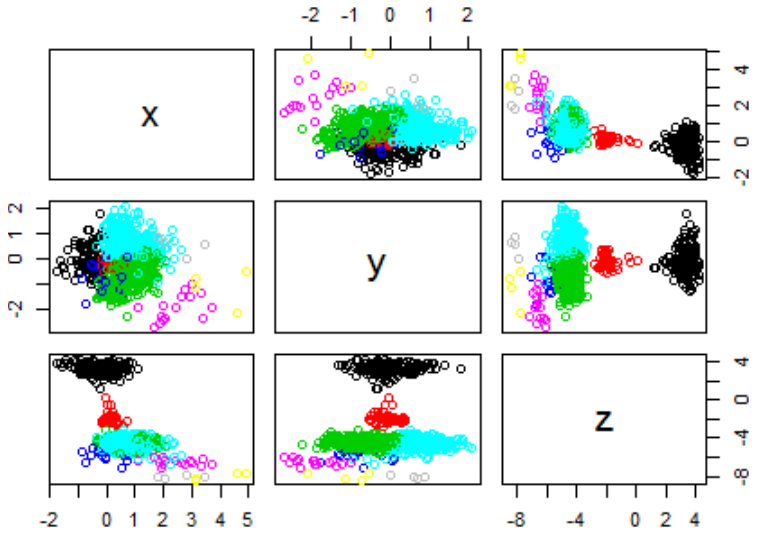
Figur B.17: Klynger opnået ved hierarkisk klyngedannelse af centeret af kolesterol-molekylet.



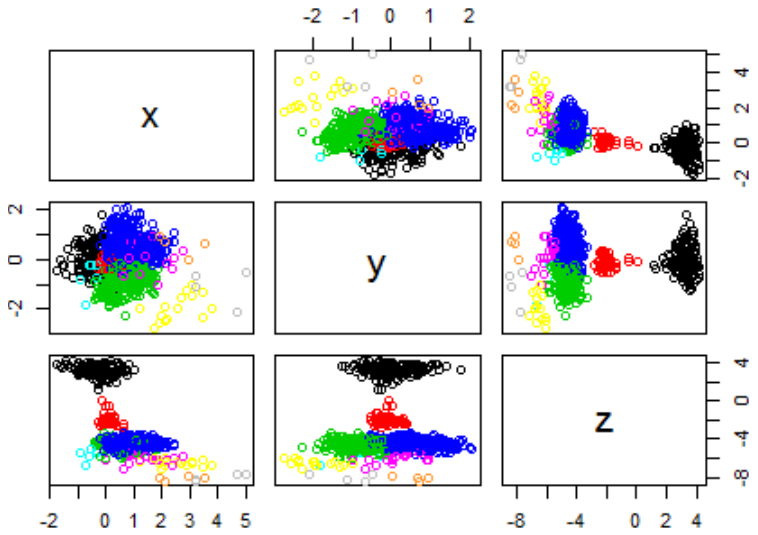
Figur B.18: Klynger opnået ved hierarkisk klyngedannelse af det massevægtede center af kolesterol-molekylet.



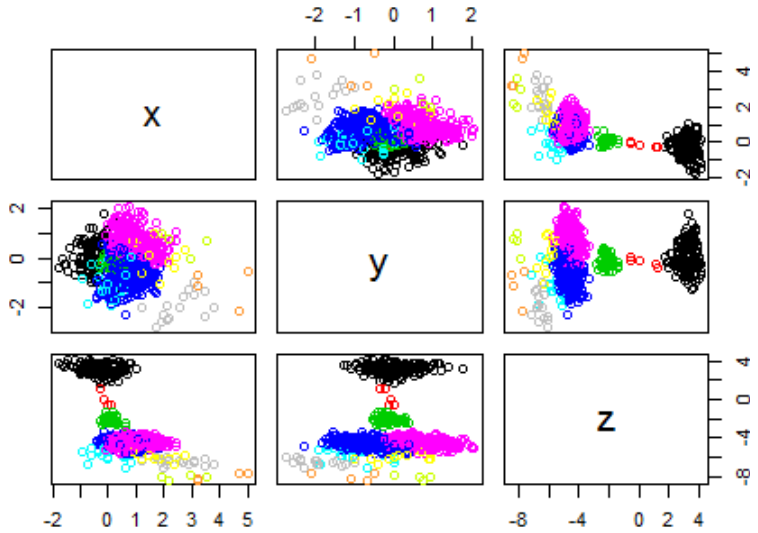
Figur B.19: Klynger opnået ved hierarkisk klyngedannelse af centeret af carbon-atomerne i kolesterol-molekylet.



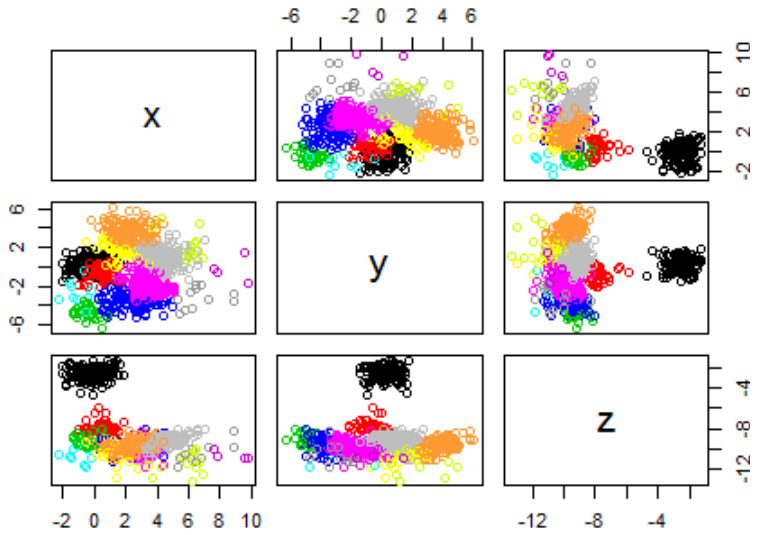
Figur B.20: Klynger opnået ved hierarkisk klyngedannelse af centeret af den rigide del af kolesterol-molekylet.



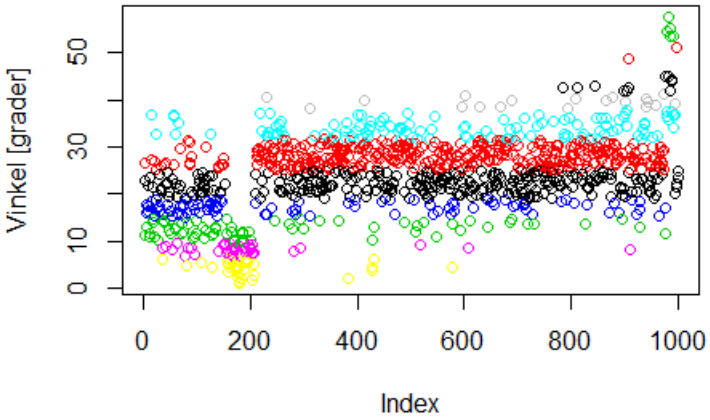
Figur B.21: Klynger opnået ved hierarkisk klyngedannelse af det massevægtede center af den rigide del af kolesterol-molekylet.



Figur B.22: Klynger opnået ved hierarkisk klyngedannelse af centeret af carbon-atomerne i den rigide del af kolesterol-molekylet.

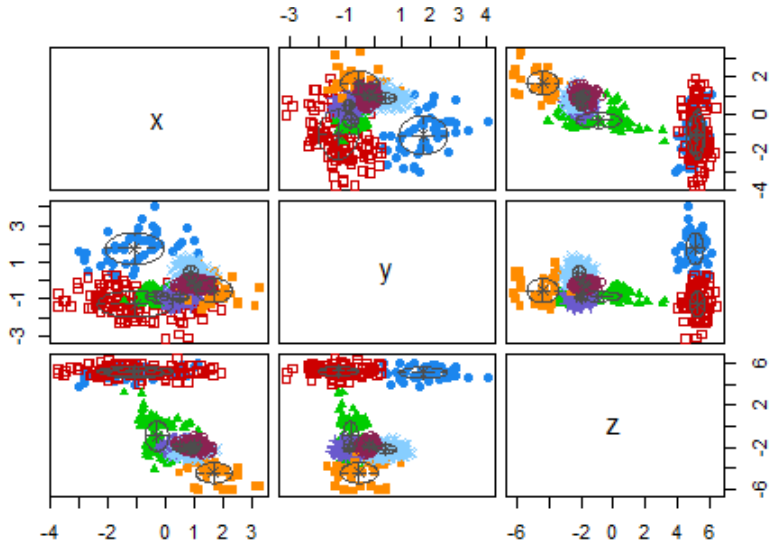


Figur B.23: Klynger opnået ved hierarkisk klyngedannelse af oxygen-atomet i kolesterol-molekylet og $K = 8$.

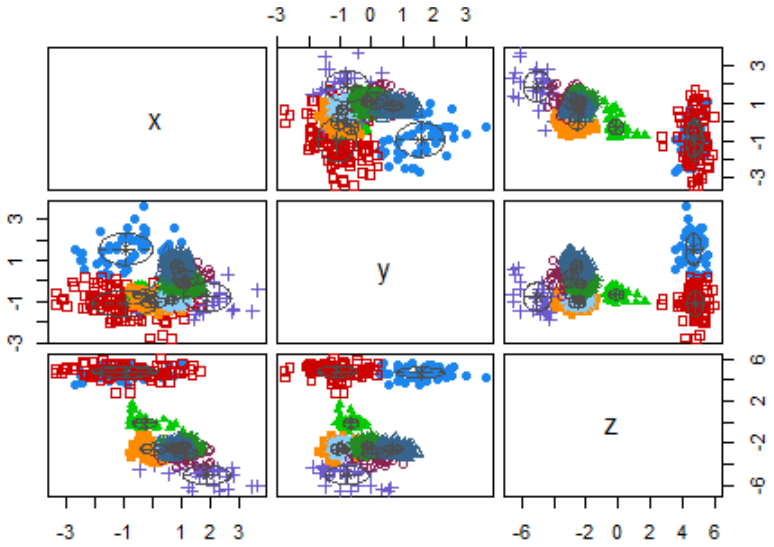


Figur B.24: Klynger opnået ved hierarkisk klyngedannelse af vinklen mellem vektoren i kolesterol-molekylet og z-aksen.

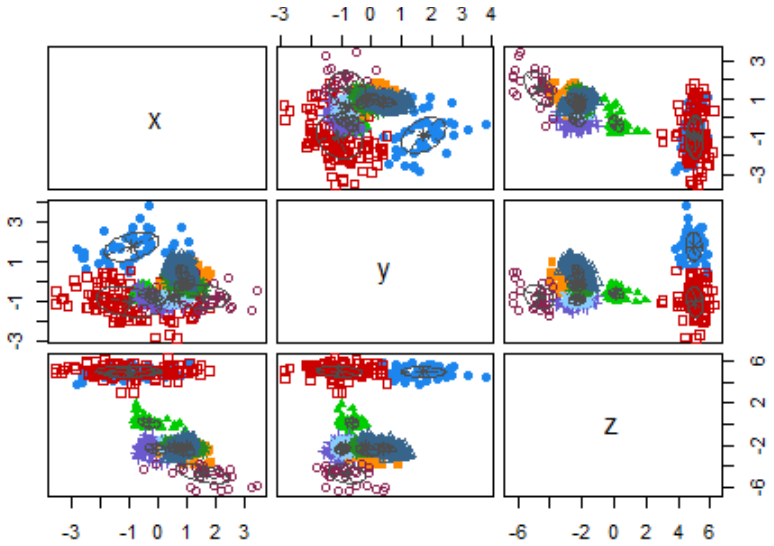
B.3 Mixture modeling



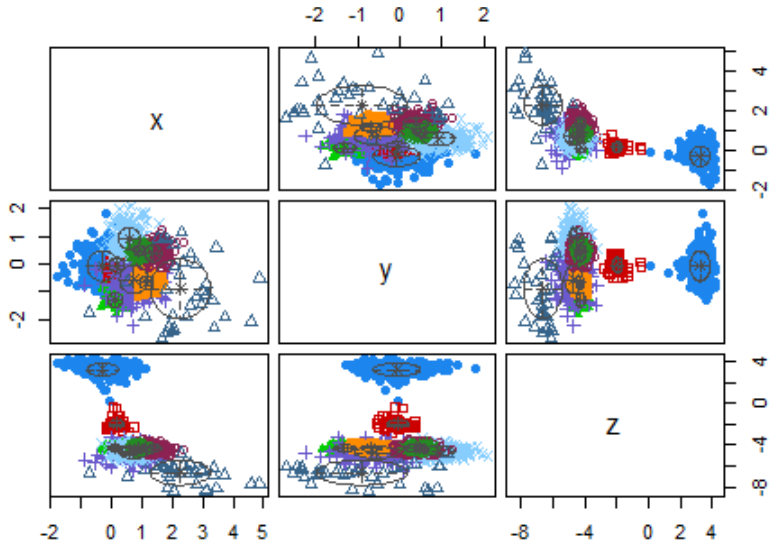
Figur B.25: Klynger opnået ved *mixture modeling* af centeret af kolesterol-molekylet.



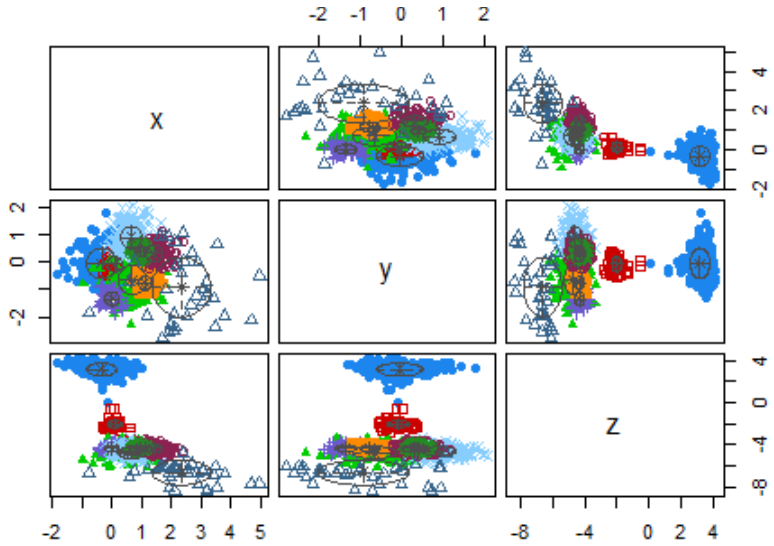
Figur B.26: Klynger opnået ved *mixture modeling* af det massevægtede center af kolesterol-molekylet.



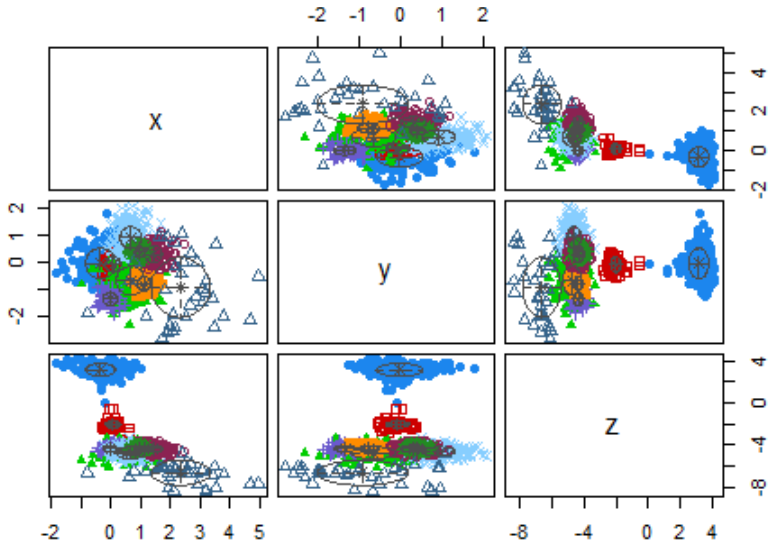
Figur B.27: Klynger opnået ved *mixture modeling* af centeret af carbon-atomerne i kolesterol-molekylet.



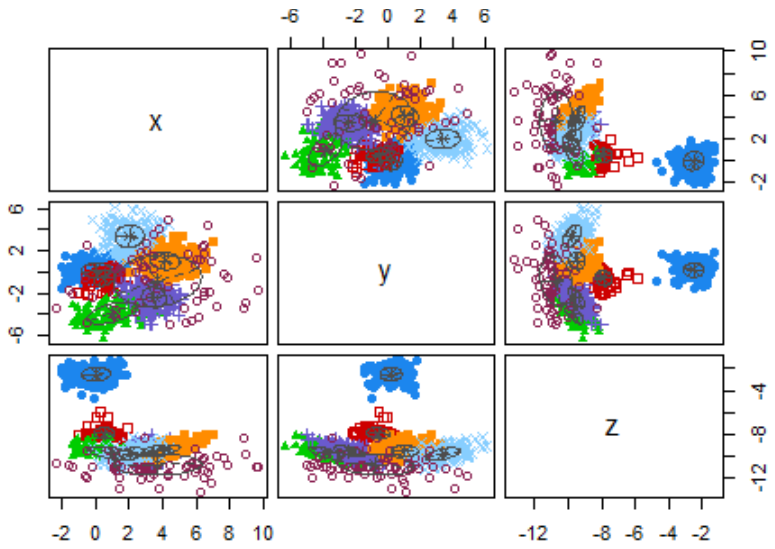
Figur B.28: Klynger opnået ved *mixture modeling* af centeret af den rigide del af kolesterol-molekylet.



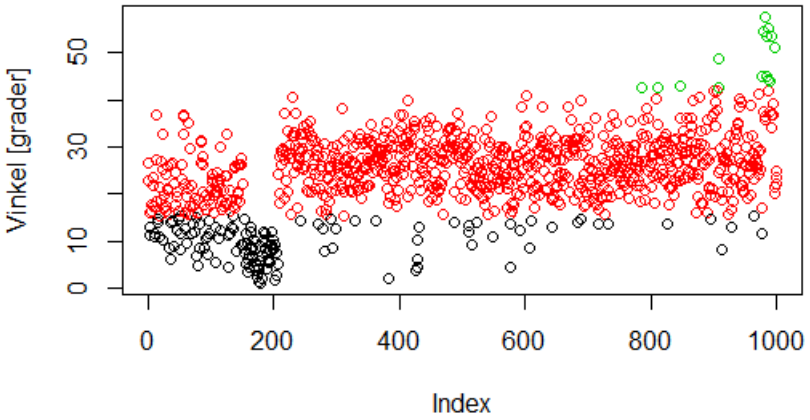
Figur B.29: Klynger opnået ved *mixture modeling* af det massevægtede center af den rigide del af kolesterol-molekylet.



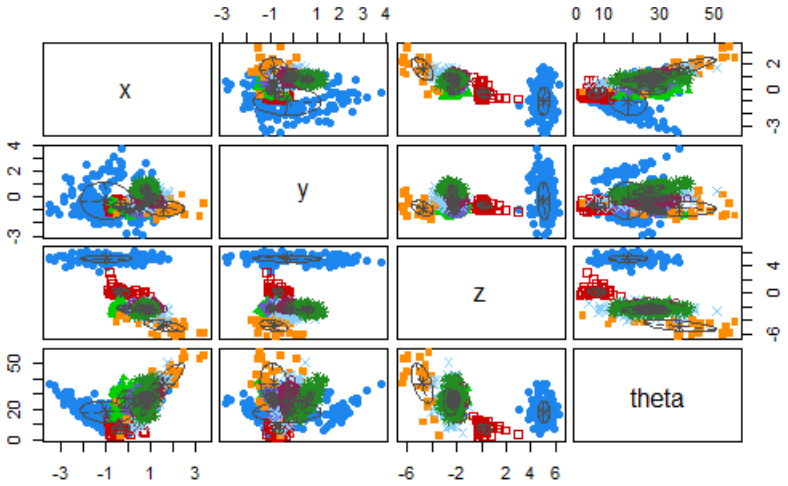
Figur B.30: Klynger opnået ved *mixture modeling* af centeret af carbon-atomerne i den rigide del af kolesterol-molekylet.



Figur B.31: Klynger opnået ved *mixture modeling* af oxygen-atomet i kolesterol-molekylet.



Figur B.32: Klynger opnået ved *mixture modeling* af vinklen mellem vektoren i kolesterol-molekylet og z-aksen.



Figur B.33: Klynger opnået ved *mixture modeling* af centeret af carbon-atomerne i kolesterol-molekylet og vinklen mellem vektoren i kolesterol-molekylet og z-aksen.