Finn årsaken til Parkinsons sykdom

Bakgrunn

I denne oppgaven er målet å finne årsaken til Parkinsons sykdom ved hjelp av ca 60 linje Python kode (vi gjør noen forenklinger og bruker litt ryddigere data enn normalt, men i virkeligheten er det ikke så mye mer enn 60 linjer kode som skal til).

Først litt rask bakgrunn: Parkinsons sykdom er en genetisk sykdom som gjør at man gradvis mister kontrollen over musklene i kroppen. At den er genetisk, betyr at det er en medfødt endring i DNA-et i cellene i kroppen som gjør at man er disponert for å få sykdommen.

Hver celle i kroppen har eksakt samme DNA-sekvens, mens DNA-sekvensen i to ulike mennesker har små forskjeller (med mindre de er eneggede tvillinger). DNA-et er en sekvens av "baseparene" A, C, G og T og kan derfor enkelt representeres som en tekst, f. eks AAATTTCTTTC osv. Man kan i dag hente ut denne sekvensen (som er ca 3 milliarder lang) fra cellen til en person.

I denne oppgaven skal vi se på DNA-sekvensen til 20 friske personer og 20 personer med Parkinsons sykdom og se om vi finner en forskjell som kan være årsaken til sykdommen. Filene vi skal jobbe med er ikke fra ekte pasienter, men kunne like gjerne ha vært det, og de inneholder samme forskjell i DNA-et som ekte pasienter ville ha hatt. Det betyr at hvis du klarer å finne denne forskjellen, vil du kunne google posisjonen du finner og få treff som indikerer at du har funnet riktig DNA-variant.

Oppgave 1: Les inn filene

Vi har DNA-sekvensen til den første millionen basepar på kromosom 4 til 20 friske og 20 syke pasienter. Last ned denne zip-filen (http://folk.uio.no/ivargry/parkinsons.zip) og pakk den ut slik at du har de 40 txt-filene som er i zip-arkivet i samme mappe som du skriver koden din.

Skriv en funksjon som leser inn innholdet fra hver fil og putter det i to lister. En liste skal inneholde sekvensene til de friske personene og en liste sekvensene til de syke pasientene. Du kan ta utgangspunkt i denne koden:

```
def les_inn_dna_sekvenser():
  friske = []
  syke = []

  for i in range(0, 20):
    nummer = str(i)
    # Les inn sekvensen til frisk person med dette nummeret her og legg til i listen friske
    # Les inn sekvensen til syk pasient med dette nummeret her og legg til i listen syke

  return friske, syke
```

```
friske, syke = les_inn_dna_sekvenser()
```

Sjekk at lengden til listen “friske” er 20 og lengden til listen “syke” også er 20, og at lengden til en av sekvensene er 1 million.

```
assert len(friske) == 20
assert len(syke) == 20
assert len(friske[0]) == 1000000
```

Oppgave 2

For hver posisjon i sekvensen (posisjon 0 til 1 million) ønsker vi å vite hvor mange som har A, C, G eller T blant de syke og hvor mange som har A, C, G, eller T blant de friske. Vi forventer at det stort sett er veldig likt, f. eks at hvis en person har A et sted, så vil de fleste andre også ha det. Det vil være små naturlige forskjeller enkelte steder.

Lag en funksjon *finn_fordeling(sekvenser)* som tar en liste med sekvenser (for eksempel listen syke), går gjennom hver posisjon i alle sekvensene i den listen, og teller opp antallet som har A, C, G og T på hver posisjon. Resultatet skal være en nestet liste hvor hvert element er en liste med 4 tall. Første tall sier antallet som har A på den posisjonen, andre tallet G, tredje tallet C og fjerde tallet T. Ta utgangspunkt i denne koden:

```
def finn_fordeling(sekvenser):
  fordeling = []

  #print(sekvenser[0])
  # Lag en tom liste med en teller for A,C,G,T på hver posisjon
  for i in range(len(sekvenser[0])):
     fordeling.append([0, 0, 0, 0])

   for i in range(0, len(sekvenser[0])):
        # i vil gå fra 0 til 1 mill. For hver posisjon i, tell opp antallet sekvenser som
        # har A, C, G og T på den posisjonen. Lagre antallet med A i posisjon 0 i listen på posisjon i,
antallet med C i posisjon 1 osv.
        # ….. fyll ut her ….

    return fordeling
```

Programmet vil etter dette se omtrent slik ut, med de to funksjonene i tillegg:

```
friske, syke = les_inn_dna_sekvenser()
friske_fordeling = finn_fordeling(friske)
syke_fordeling = finn_fordeling(syke)
```

friske_fordeling og *syke_fordeling* vil være to nestede lister. Hvis du printer de første 10 elementene fra *friske_fordeling* med koden *print(friske_fordeling[0:10])* skal det se slik ut:

[[0, 0, 20, 0], [20, 0, 0, 0], [0, 20, 0, 0], [0, 0, 20, 0], [0, 0, 20, 0], [20, 0, 0, 0], [0, 0, 20, 0], [0, 0, 20, 0], [0, 0, 0, 20], [20, 0, 0, 0]]

Oppgave 3: Finn forskjellen

Vi har to nestede lister som gir fordelingen av A, C, G og T på hver posisjon i kromosom 4 for syke og friske personer. Vi ønsker nå å finne ut hvor det er stor forskjell i fordelingen (f. eks at syke personer stort sett har en A mens syke personer har en C).

Skriv en funksjon som tar inn de to listene som holder på fordelingene og går gjennom hvert element og printer ut hver gang de to fordelingene på en posisjon er ulike.

```
def finn_forskjeller(friske_fordeling, syke_fordeling):
  for i in range(0, len(friske_fordeling)):
    # Sjekk om friske_fordeling og syke_fordeling på denne posisjonen er ulike og print ut posisjonen i
    # Tips: du kan sjekke om to lister er like ved å bare skrive “if liste1 == liste2”
```

Du bør få ut ca 30 posisjoner hvor de to fordelingene er ulike. De fleste stedene vil det bare være små forskjeller (f. eks en eller to personer som har noe annet enn det som er normalt). Men på én av disse posisjonene vil det være en stor forskjell mellom hva syke og friske personer har. Finn ut hvilken posisjon dette er manuelt ved å se gjennom det du printer ut, eller ved å skrive kode som luker ut store forskjeller

Når du har funnet en posisjon i, så kan du google **parkinson "4:posisjonen"** (4 er fordi vi er på kromosom 4). Pluss på 1 på posisjonen først, ettersom den trolig er 0-indeksert (mens man vanligvis starter genomet på 1). Hvis du har funnet riktig posisjon, så bør du få et treff når du googler.