Den genetiska koden

Den genetisk koden består som bekant av kodoner, dvs bastrippletter som kodar för en aminosyra som sedan kommer att ingå i en proteinkedja. Numera, när det finns effektiva metoder för att läsa innehållet i DNA sekvenser, så finns det mycket data inom biostatistik. Ett sådant dataset som ni ska analysera består av mutationsfrekvenser mellan kodoner och är baserat på sju miljoner bastrippletter från ett antal olika djurarter. Ni har fått en matris som sammanfattar data genom att ge sannolikheten att en kodon muterar till en annan kodon (sannolikhet per tidsenhet), tex $P_{AAC \text{ --> } AAA}$.

I den genetiska koden finns det stor redundans, dvs flera kodtrippletter kodar för samma aminosyra. Det finns $4^3 = 64$ kodtrippletter men dessa kodar bara för 21 aminosyror . Redundans är intressant och dess struktur har diskuterats flitigt i litteraturen (debatten är inte slut ännu). En sak som man lätt kan konstatera är att koden är evolverad för att vara robust, dvs in punktmutation av en bas har mindre sannolikhet att byta aminosyra än man skulle förvänta sig av en slumpmässig kod.

Tänk er nu att ni har lyckats mäta upp mutationsfrekvenserna men i övrigt vet mycket lite om den bakomliggande koden. Men ni misstänker att det kan finnas redundans i mappningen från kod till funktion. Denna redundans bör då synas i era data genom att det är mer sannolikt att observera neutrala mutationer som inte ändrar aminosyra. Hypotes baseras naturligtvis på att de flesta mutationer är skadliga och att det därför i regel är dåligt att byta ut en aminosyra. Frågan är nu hur ni ska gå till väga för att hitta neutrala grupper av kodons som mappar till samma aminosyra, dvs ni ska blint identifiera den genetiska koden.

Detta problem kan lösas genom att vi identifierar metastabila tillstånd i markovmatrisen (P) som vi mätt upp (laddat ner). Som vi diskuterat på lektionen så kan detta lösas som ett minimeringsproblem:

$$\min_{G_{ij}} (P_{ij} - \beta E_{ij}) G_{ij}$$

där matrisen G markerar att kodon i och j är i samma grupp genom att sätta Gij = 0, annars är Gij = 1, E är en matris med bara ettor som element och beta är en parameter som gör att man undviker det triviala mimimat där alla kodon är i var sin grupp. Som vi också diskuterade på lektionen så är minimeringsproblemet lättare att hantera om man tänker sig en binär splitt till två grupper och då inför hjälpvariabler (spinn för att låna fysiktermer):

$$G_{ij} = 1 - s_i s_j$$

där nu s_i är ±1. Detta leder till följande minimeringsproblem

$$\min_{s_i \in \{-1,1\}} -s_i (P_{ij} - \beta E_{ij}) s_j$$

som kan lösas på flera olika sätt. I ert fall bör det räcka med en “greedy algorithm”, dvs starta med slumptilldelning i vektorn s, ändra slumpvisa element i s och acceptera om det leder till förbättring av värdefunktionen.Värdet på parameterna beta kan man experimentera med men en bra utgångspunkt är beta = 1/antalet kodoner.

Minimeringen i spinn-version delar bara kodonerna i två grupper. För att få en total uppdelning måste ni använda algoritmen rekursivt (eller på annat sätt successivt fortsätta dela grupperna tills minimeringen inte längre föreslår någon delning.

Resultatet bör avslöja den genetiska koden. Notera att den “greedy algorithm” som jag föreslår inte alls är den bästa för att lösa problemet, men den är lätt att implemetera och bör räcka i detta fallet. Ni bör köra hela proceduren ett par gånger och ta till vara det bästa resultatet.

För att ni lättare ska komma igång så har ni också fått två enklare testmatriser. Använd dessa för att utveckla era algoritmer.

Tips: metastabila tillstånd i en markovmatris syns tydligt i en matrisplott om rader och kolumner är sorterade så att tillstånd i samma metatillstånd ligger bredvid varandra.

Kommentar: matrisen ni fått har bara 61 kodons, detta beror på att jag tagit bort start och stopp koderna för att underlätta för er.