Stofnerfðagreining með skjákortum

Matthías Páll Gissurarson*†

31. maí 2017

Útdráttur

Stofnerfðagreining með skjákortum verkefnið snýst um að kanna hvort hægt sé að hraða á útreikningum í stofnerfðagreiningarforritinu `structure` með því að notast við venjuleg skjákort til þess að framkvæma útreikningana. Þá er hugmyndin að vísindamenn sem stunda rannsóknir í lífvísindum þar sem á stofnerfðagreiningu þarf að halda geti hraðað reikningum með því að nýta innbyggð skjákort á almenningstölvum.

Inngangur

Hægt er að ná verulega auknum hraða á forritum með því að færa þau yfir á skjákort, ef þau eru þess eðlis að sömu aðgerðunum er verið að beita á mismunandi gögn.

Skjákort hafa þróast frá einföldum tækjum til að koma stafrænum myndum á skjá yfir í öflug reiknitæki. Sífellt auknar kröfur í þrívíddartölvuleikjum leiddu til þróunar þar sem skjákort inniheldur í raun hundruð til þúsundir lítilla örgjörva með sæmilega reiknigetu. Á síðastliðnum árum hafa komið fram forritunarumhverfi sem gera okkur kleift að skrifa forrit fyrir skjákort.

Þetta krefst þó endurskoðunar á þeim reikniritum sem liggja að baki. Mörg þeirra eru hönnuð með aðeins einn örgjörva í huga, þ.e. að aðeins sé hægt að gera einn hlut í einu. Sérstaklega þarf að hafa í huga að ekki sé verið að breyta sömu hlutunum

*Leiðbeinandi Páll Melsted

†Styrkt af Nýsköpunarsjóði námsmanna og Háskóla Íslands

á sama tíma og hvernig á að skipta verkum þannig að sem flestir örgjörvar geti verið að vinna að verkefninu í einu.

Að auki þarf að endurforrita frá grunni til að nýta það reikniafl sem skjákort bjóða upp á.

Rannsóknin felst í því að athuga hvort hægt sé að hraða á forritinu `structure` [6][4], stofnerfðagreiningarforriti eftir Pritchard et. al., með því að endurskrifa það til þess að það noti skjákort við útreikninga.

`Structure` er stofnerfðagreiningar forrit sem tekur inn upplýsingar um hlutföll genasamsæta (allele) á ákveðnum staðsetningum á litningi (loci) hjá einstaklingum og fjölda stofna K. `structure` beytir svo `MCMC` aðferð til þess að finna líklegustu skiptingum einstaklingana upp í K stofna.

`Structure` vinnur aðalega með þrjár breytur, X, P, Z og Q, þar sem X, P, Z og Q eru vigrar. X táknar hlutföll genasamsæta á ákveðnum staðsetningum á litningi hjá einstaklingum. Z táknar í hvaða stofni hver einstaklingur er, P táknar hlutföll á genasamsætum í hverjum stofni og Q táknar hve hátt hlutfall af genasamsætum einstaklings kemur frá hverjum stofni.

`MCMC` (Markov Chain Monte Carlo) aðferð [5] byggir á að búa til Markov keðju sem hefur sömu dreifingu og líkindadreifingin sem á að herma, en í okkar tilfelli erum við að herma $\Pr(P, Z, Q|X)$. Þetta má gera með því að herma $\Pr(P|Z, Q, X)$, $\Pr(Q|P, Z, X)$ og $\Pr(Z|Q, P, X)$ [6].

`Structure` hefur tvær útgáfur af hermun. Einfaldari útgáfan gerir ráð fyrir að einstaklingar séu aðeins úr einum stofni, en flóknari útgáfan leyfir einstaklingum að vera úr fleiri en einum stofni.

Í upphafi byrjar `structure` á því að skipa hverjum einstaklingi í stofn af handahófi, með því að gera ráð fyrir að þeir séu jafn líklegir til þess að vera úr hverjum stofni.

Í einfaldari útgáfunni er hver ítrun `structure` eftirfarandi:

`Structure` hermir líklegt P, hlutföll genasamsæta í hverjum stofni, með því að gera ráð fyrir að stofnarnir séu í raun og veru eins og núverandi skipun einstaklinga í stofna segir til um, s.s. metur P útfrá X og núverandi Z.

`Structure` hermir svo líklegt Z, í hvaða stofni hver einstaklingur er, með því að gera ráð fyrir að hlutföll genasamsæta í hverjum stofni er eins og P segir til um, s.s. metur Z útfrá X og núverandi P.

Í flóknari útgáfunni hermir `structure` bæði P og Q í fyrra skrefinu og hermir svo Z útfrá bæði P og Q í seinna skrefinu.

Aðferð

Tæki

Við gerð rannsóknarinnar var notast við tölvu með `Ubuntu 14.04LTS` og `Nvidia` skjákort af gerðinni `Nvidia 760 GTX`.

Forritunin fór fram í `OpenCL`, en það er opinn staðall sem flestir skjákortaframleiðendur styðja, en þó misvel.

Framkvæmd

Rannsóknin fór þannig fram að kóði `structure` forritsins var endurskrifaður og nær allar aðgerðir færðar yfir á skjákort.

Þetta var gert með því að athuga hvernig hægt væri að skipta aðgerðunum upp til þess að sem flestar aðgerðir gætu verið gerðar samtímis.

Slembigjafi

Til þess að geta framkvæmt hermanirnar á skjákortinu þarf að vera með slembigjafa. Slembigjafar eru ekki innbyggðir á flestum skjákortum og því þarf að útfæra gervislembigjafa til þess að nota, sem er nógu góður til þess að hann dugi í útreikningunum.

Upprunalega var slembigjafinn `mwc64x` fyrir valinni og honum bætt við í forritið. Síðar kom í ljós að sá slembigjafi virkaði ekki á skjákortum frá `AMD`, og því þurfti að skipta um slembigjafa. Þá varð `GGL` slembigjafinn [3] fyrir valinu, en hann var talinn einfaldastur en þó nógu góður til þess að duga í verkefnið.

Hermun

Þegar slembigjafinn var kominn var hægt að útfæra hermanirnar á skjákortinu, en til þess þurfti að færa hin ýmsu líkindaföll yfir á skjákortið.

Þá var passað að hægt væri að framkvæma sérhverja ítrun algerlega á skjákortinu. Þetta var gert til þess að ekki þyrfti að flytja gögnin frá skjákortinu yfir á örgjörvan og tilbaka í hverri ítrun, því færsla á gögnum milli skjákorts og örgjörva tekur mikinn tíma.

Þá framkvæmir forritið eitt skref í einu, en það byrjar á að herma Z, svo hermir það P og svo loks Q.

Til þess að nýta skjákortið sem best þurfti að endurhugsa skrefin frá því að vera að nota lykkjur og gera fyrst eitt og svo annað yfir í að vera að gera margt í einu. Til þess að einfalda okkur þetta voru skrefin endurhugsuð sem `map` og `reduce` aðgerðir.

`map` og `reduce` aðgerðir eru mikið notaðar til þess að einfalda samhliða útreikninga, en `map` aðgerðin byggir á að beita sömu aðgerð á öll gögnin, og `reduce` aðgerðin tekur svo saman niðurstöðurnar.

Þannig var `UpdateZ` fallið til dæmis bara ein `map` aðgerð, sem gat þá hermt hvern einstakling samhliða.

í `UpdateP` er P hermt útfrá núverandi Z (með millibreytum Epsilon og Fst), og nýtt P búið til úr Dirichlet dreifingu með stikum útfrá breytunum Epsilon og Fst.

Í `UpdateQ` er Q hermt með því að fyrst búa til nýtt gildi fyrir hvert núverandi gildi í Q. Síðan eru reiknaðar milliniðurstöður sem segja til um líkurnar á fyrir hvern einstakling að nýja Q-ið fyrir þann einstakling passi við einstaklinginn miðað við í hvaða stofni hann er og hvernig hlutföll genasamæta eru hjá honum.

Að lokum er svo próf sem samþykkir fyrir hvern einstakling með líkum háðum hversu líklegt er að nýja Q-ið sé fyrir einstaklinginn. Ef það er samþykkt, þá er Q-ið fyrir þann einstakling uppfært, annars ekki.

Þetta er svo endurtekið þar til að tilsettum fjölda ítrana hefur verið náð, en þá er lokastaðan athuguð og sagt til í hvaða stofni einstaklingar lentu.

Prófanir

Samhliða þróun voru framkvæmdar prufukeyrslur á þekktum gögnum til þess að athuga hvort niðurstöður sem forritið skilað væru ekki í samræmi fyrir og eftir að forritið var fært yfir á skjákort.

Niðurstöður

Forritið keyrir nú algjörlega á skjákortinu, með virkni fyrir helstu valmöguleikana sem er í boði á upprunalega forritinu. Þá var miðað við þá virkni sem er sjálfgefin í forritinu og hún útfærð fyrst, en síðar var bætt við virkni sem er ekki sjálfgefin. Enn á eftir að útfæra suma virkni sem er ekki sjálfgefin, en áhersla var lögð á að sú virkni sem mest er notuð væri útfærð fyrst.

Þegar forritið var keyrt á skjákortinu, þá var það aðeins verra en upprunalega forritið fyrir lítil gögn.

Tímamælingarnar gáfu eftirfarandi fyrir 30,000 ítranir

	`structure`	Skjákorts `structure`
Lítil gögn	5 sek	8 sek
Stór gögn	1 klst 23 mín	19 mín

Litlu gögnin voru fengin frá höfundum forritsins, en það eru gögn sem eru tekin sem dæmi um gögn sem nota má í `structure`. Þau gögn eru hermd út frá gefnum stofnum, en svo er `structure` notað til þess að finna út hverjir gefnu stofnarnir voru. Þannig er hægt að nota litlu gögnin og vera viss um að forritið sé að skila réttri niðurstöðu. Litlu gögnin eru 200 einstaklingar með 5 staðsetningum, en skráin er 7.5 Kb að stærð.

Stóru gögnin eru gögn sem notuð voru í pappírnum "A worldwide survey of haploype variation and linkage disequilibrium in the human genome" [2], en þau eru fáanleg frá vefsíðu höfundar `structure`, og eru gögnin á `structure` formi, þ.e. formi sem passar inn í `structure`. Stóru gögnin eru 927 einstaklingar með 2834 staðsetningum, en skráin er 11 Mb að stærð.

Umræða

Tilgáta okkar um hraðaaukningu vegna skjákorts reyndist vera rétt. Engu að síður þyrfti að huga betur að hvort niðurstöðurnar sem fást þegar þetta er keyrt á skjákorti séu þær sömu og þegar upprunalega forritið er keyrt.

Það var áhugavert að sjá afl `OpenCL`, en það kom þó í ljós þegar á leið rannsóknina að þau forritunartól sem eru í boði fyrir `OpenCL` eru ekki nógu góð. Því mætti frekar nota `CUDA` við álíka verkefni næst, en það er forritunarmál frá `Nvidia` sem hefur verið lengur í þróun og nýtur mun betri stuðnings. Helsti galli `CUDA` er þó að það keyrir bara á `Nvidia` skjákortum, og því gætu notendur sem ekki eru með `Nvidia` kort ekki keyrt forritið. Ofurtölvur sem nýta skjákort eru nær eingöngu með `Nvidia` kort en í almenningstölvum er skiptingin jafnari milli `Nvidia`, `AMD` og `Intel` skjákorta.

Hröðunin passar við niðurstöður úr fyrri rannsóknum [1] um að skjákort geti hraðað á útreikningum á álíka forritum, en `structure` notast við `MCMC` fyrir stofngreiningar.

Það fór úrskeiðis í framkvæmd að athuga ekki nægilega vel muninn á keyrslunni þegar komið var út í flóknari slembiföll, og gat það valdið gífurlegri reikniskekkju.

Það er þó hægt að sjá hvenær þessi reikniskekkja kemur upp, en við prófanir kom í ljós að reikniskekkjan kom upp í 76 af 667 keyrslum, en það eru um 11% keyrslna.

Villan virðist koma upp vegna hvernig skjákortið skipuleggur útreikningana, og því hefur reynst erfitt að koma í veg fyrir skekkjuna og finna ástæður hennar. Við höfum þó fundið út hvaðan í forritinu villan kemur, og því er hægt að koma í veg fyrir hana með því að láta örgjörvann um hluta vinnslunar, en sú lausn dregur úr hraða forritsins.

Þrátt fyrir margar tilraunir við að bæta úr þeirri skekkju er hún enn til staðar í forritinu núna. Einnig var forritið ekki prófað nógu vel á öðrum skjákortum en `Nvidia` kortinu, og því kom í ljós að þegar átti að keyra þetta á öðrum skjákortum að villur komu upp við þýðingu og keyrslu. Þegar ráðið hafði verið úr þeim kom í ljós að munur var á milli keyrslna á `Nvidia` skjákortinu og nýja skjákortinu.

Framtíðarplön

Vinna við skjákorta `structure` mun halda áfram, en þá er markmiðið að laga villurnar og tryggja að reikniskekkjan komi ekki upp.

Þá er áætlunin að gefa forritið út og skrifa stuttan pappír um virkni forritsins.

Allur kóði er fáanlegur á `https://github.com/Tritlo/structure`, en eftir á að finna leyfi á kóðann.

Heimildir

[1] Shuai Che, Michael Boyer, Jiayuan Meng, David Tarjan, Jeremy W Sheaffer, and Kevin Skadron. A performance study of general-purpose applications on graphics processors using cuda. *Journal of parallel and distributed computing*, 68(10):1370–1380, 2008.

[2] Donald F Conrad, Mattias Jakobsson, Graham Coop, Xiaoquan Wen, Jeffrey D Wall, Noah A Rosenberg, and Jonathan K Pritchard. A worldwide survey of haplotype variation and linkage disequilibrium in the human genome. *Nature genetics*, 38(11):1251–1260, 2006.

[3] Vadim Demchik. Pseudo-random number generators for monte carlo simulations on ati graphics processing units. *Computer Physics Communications*, 182(3):692–705, 2011.

[4] Daniel Falush, Matthew Stephens, and Jonathan K Pritchard. Inference of population structure using multilocus genotype data: linked loci and correlated allele frequencies. *Genetics*, 164(4):1567–1587, 2003.

[5] WR Gilks, S Richardson, and David Spiegelhalter. *Markov Chain Monte Carlo in Practice*. CRC Press, 1995.

[6] Jonathan K Pritchard, Matthew Stephens, and Peter Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, 2000.