BIOLOOGIA-GEOGRAAFIA TEADUSKOND
MOLEKULAAR- JA RAKUBIOLOOGIA INSTITUUT
BIOINFORMAATIKA ÕPPETOOL

Meelis Olev

PCR praimerite automaatne grupeerimine

Keskastme lõputöö

Juhendaja: Maido Remm, Prof., PhD

Tartu, 2003-2004

Sisukord

Lühendid ja mõisted....4
Sissejuhatus....5
I Teoreetiline osa....6
1 PCR....6
1.1 PCR praimerite disain....6
1.2 PCR praimerite disainiks kasutatavad programmid....7
2 PCR kvaliteeti mõjutavad parameetrid....8
2.1 Praimerite sulamistemperatuur....8
2.2 Kordusjärjestused....8
2.3 Polü-N traktid....8
2.4 Praimerite vahelised interaktsioonid....9
2.5 Produkti pikkus....9
2.6 Praimerite pikkus....9
2.7 Praimerite GC nukleotiidide sisaldus....10
3 Seostumise arvutamise meetodid....10
3.1 Aukudeta joondus....10
3.2 Smith-Waterman....10
3.3 BLAST....11
3.4 Suffiksipuud....11
3.5 Räsialgoritmid....12
3.6 Kui pikka ala praimerist arvestada....12
4 PCR multipleks....13
4.1 Olulised parameetrid grupeerimisel....13
4.1.1 Energia piirväärtused (cutoffs)....13
4.1.2 Produktide pikkuse erinevuse piirväärtused....13
4.1.3 Praimerite sulamistemperatuuride erinevus....13
4.1.4 Piirangud grupi suurusele....14
4.2 Grupeerimise kvaliteeti mõjutavad tegurid....14
5 Alternatiivid praimerite grupeerimisele....14
5.1 PCR supressioonil baseeruv amplifitseerimine (Broude et al., 2001)....14
5.2 DOP-PCR (Jordan et. al, 2001)....15
II Praktiline osa....16
EESMÄRK....16
1 Programmi Multiplx üldine kirjeldus....16
2 Parameetrite ja käskude kirjeldus....16
2.1 Käsud....16
2.2 Võtmed....18
3 Skoori arvutamise meetodid....18
3.1 Lokaalne joondus....18
3.2 Globaalne joondus....19
3.3 Lokaalne „lõppjoondus“ (3'-lõpu joondus)....19
4 Grupeerimismetoodika....19

5 Tulemused (proof of principle)....22
5.1 Programmi töökiirus....22
5.2 Programmi mäluvajadus....22
5.3 Programm praktilises kasutuses....23
Arutelu....24
Kokkuvõte....25
Automated grouping of PCR primers....26
Viited....27
Lisad....30

Lühendid ja mõisted

auk	joonduses mittekattuvate nukleotiidide piirkond (*gap*)
matriits	ahel, mille pealt DNA polümeraas uue, komplementaarse ahela sünteesib (*template*)
PCR	polümeraasahelreaktsioon (*polymerase chain reaction*)
piirav väärtus	väärtus, millega leitud skoori võrreldakse, üldjuhul ei ole piirava väärtuse ületamine lubatud (*cutoff*)
skoor	väärtus, sobivuse hinnang (*score*)
SNP	ühenukleotiidne polümorfism (*single nucleotide polymorphism*)

Sissejuhatus

Nii nagu tundub tänapäeval kummaline kohata bioloogi, kes ei tea midagi geenidest, ei saa ilmselt tõsiselt võtta ka ühtegi biotehnoloogiaga tegelevat inimest, kellele midagi ei ütle tähekombinatsioon PCR, eestikeelse vastega polümeraasahelreaktsioon. PCR abil on võimalik paljundada valitud lõike genoomsest materjalist (eeldusel, et sellised lõigud seal eksisteerivad), et hiljem teha need näiteks nähtavaks elektroforeesil. Meetod on laialt kasutuses mitmete genoomse materjaliga seotud toimingute juures, muuhulgas ka inimese vereproovist eraldatud genoomsete elementide analüüsiks. Tüüpilises PCR reaktsioonis osalevad polümeraas (kuumaveebakterist saadud genoomset materjali paljundav valk), genoomne materjal, kaks praimerit (oligonukleotiidid, mis seonduvad paljundatava järjestuse otstele) ja nn. PCR puhver. Inimese geenide analüüs vereproovist on tüüpiline näide, kus soovitakse sama genoomse materjali pealt paljundada e. amplifitseerida mitmeid erinevaid lõike. See nõuab kas rohkemate praimerite reaktsioonisegusse lisamist või identse genoomse materjali piisava hulga korral iga lõigu jaoks eraldi reaktsiooni läbiviimist. Inimese vereproovide puhul ei ole mõtet materjaliga pillavalt ümber käia ning seetõttu on mõttekam eri lõike korraga amplifitseerida, samas tuleb sellisel juhul arvestada praimerite omavahelist või produktidega seondumist ning mitmeid teisi piiravaid tegureid. Selliseid tegurid seavad piirangu korraga amplifitseeritavate genoomilõikude hulgale, samuti piirab ka kasutatav metoodika. Uuritavad piirkonnad ning neile vastavad praimerid tuleks jaotada gruppidesse nii, et grupid ei oleks liiga suured ega liiga väikesed ning samas oleks kasvõi genoomse materjali säästmise huvides kasulik, kui gruppe oleks võimalikult vähe. Antud probleemile lahenduse leidmiseks sai välja töötatud arvutiprogramm nimega Multiplx (miks mitte Multiplex – põhjus on autori teatavas nostalgias MS-DOSi-laadsete süsteemide suhtes, kus teatavasti failinime pikkus ei saanud olla üle 8 märgi ning Multiplex oleks samas liiga triviaalne lähenemine programmi nimetamisel), sellele programmile ongi pühendatud antud töö. Programm ise ei ole kahjuks vabalt kättesaadav, õigused sellele on ostnud üks Tartu biotehnoloogiafirma, kes müüb antud programmi edasiarendatud, Windows-platvormil töötavat versiooni WinMultiplx.

I Teoreetiline osa

1 PCR

PCR (Polymerase Chain Reaction e. polümeraasi ahelreaktsioon) on molekulaarbioloogias laialt rakendatav meetod, mida võib kasutada genotüpiseerimisel, punktmutatsioonide uurimisel, genoomse DNA kloneerimisel jm. Eesmärk on saada ühest geenilõigust palju koopiaid, mida tänu suurele hulgale on näiteks hiljem lihtsam detekteerida. PCR koosneb kolmest etapist, mida korratakse tsükliliselt:

- denaturatsioon – genoomse DNA puhul temperatuuril 93-95 °C. DNA kaksikahel lõhutakse kaheks üksikahelaks. Sellisel temperatuuril lõpevad ära ensümaatilised reaktsioonid, näiteks DNA süntees eelmisest tsüklist.
- annealing e. praimerite seostumine DNA-le (~50-70 °C). Üheahelalise praimeri ja üheahelalise matriitsi (*template*) vahel moodustuvad ja katkevad vesiniksidemed. Stabiilsemale kaksikahelale saab seostuda polümeraas (tüüpiliselt kuumaveebakterist eraldatud nn. Taq-polümeraas) ning algab matriitsi duplitseerimine.
- ekstensioon e. DNA süntees (70-75 °C). Polümeraas lisab dNTP-sid praimeri 3' otsa. Praimerid, mida polümeraas on juba pikendanud, on stabiilsemad tänu tekkinud uutele vesiniksidemetele, ülejäänud praimerid matriitsi küljes ei püsi ning võivad seonduda teistele positsioonidele järgmises tsüklis.

Tsüklit korratakse 25-45 korda (Glick, Pasternack, 1998).

1.1 PCR praimerite disain

PCR teostamiseks on vaja teada amplifitseeritava DNA järjestust, mille alusel disainida praimerid – 15-30-meersed sünteetilised nukleotiidid. Praimerite disainil tuleb arvestada järgnevaga:

- praimerite sulamistemperatuur (T_m) peaks olema 50-70 °C
- praimerite GC sisaldus peaks olema 50-55%, väiksema GC sisalduse puhul tuleb sulamistemperatuuri tõstmiseks üle 50 °C disainida praimerid pikkusega üle 20 nukleotiidi

- vältida tuleks ühe või kahe nukleotiidi kordusi, eriti 3’ otsas, kuna see võib põhjustada sekundaarseid hübridisatsioone
- praimeri 3’ ots on stabiilsem, kui seal asub G või C nukleotiid
- praimerid ei tohiks moodustada sekundaarstruktuure ning di- ja oligomeere

1.2 PCR praimerite disainiks kasutatavad programmid

Suured molekulaarbioloogias kasutavad programmide paketid nagu EMBOSS (http://www.hgmp.mrc.ac.uk/Software/EMBOSS/, vabavara) ja GCG (http://www.accelrys.com/about/gcg.html, kommertstarkvara) sisaldavad programme praimerite disainimiseks, vastavalt Prima ja Prime. Laialt tuntud ning ka veebibrauseri kaudu kasutav on Primer3 (Rozen, Skaletsky, http://frodo.wi.mit.edu/primer3/primer3_code.html, vabavara) (veebi kaudu kasutatav variant http://fokker.wi.mit.edu/cgi-bin/primer3/primer3_www.cgi), varasemates versioonis nimega Primer. Erinevate tööpõhimõtete ja sisendandmetega on ka näiteks programmid PrimerDesign (http://www.chemie.uni-marburg.de/~becker/pdhome.html, DOS), PC-Rare (Griffais, http://bip.weizmann.ac.il/mb/bioguide/pcr/PCRsofPC_Rare.html, Mac, Windows), PRIDE (http://www.uni-duesseldorf.de/urz/hardware/server/pride/PRIDE.html, Solaris, Digital Unix, Linux), PerlPrimer (http://perlprimer.sourceforge.net/, Windows, Unix, Mac). Veebibrauseri kaudu kasutatavad on veel NetPrimer (http://www.premierbiosoft.com/netprimer/netprlaunch/netprlaunch.html, Windows, Mac), GeneFisher (http://www.genefisher.de/, CGI), Web Primer (SGD, http://seq.yeastgenome.org/cgi-bin/web-primer, CGI). Toodud nimekirja võiks veel pikalt jätkata.

Optimaalse praimeripaari leidmine nõuab mitmete parameetrite optimeerimist. Viimase kümne aasta jooksul on kujunenud välja kindel hulk põhiparameetreid, mida kasutavad paljud programmid (Burpo, 2001). Neist on täpsemalt juttu allpool. Enamik programme leiavad praimeri kriteeriumite tegelikud väärtused ning suunavad neid, et leida optimaalne praimer kindlaks kasutusjuhtumiks (Kampke, et al, 2001). Kuid näiteks PRIDE kasutab kvalitatiivset kasutaja sisendit ning arvutustes hägusat loogikat (Haas, et al, 1998).

2 PCR kvaliteeti mõjutavad parameetrid

PCR kvaliteedi all tuleb mõista põhiliselt seda, kuivõrd efektiivselt seostuvad praimerid õigetesse kohtadesse ning seostumise stabiilsust. Kvaliteedi määravad põhiliselt järgnevalt toodud parameetrid.

2.1 Praimerite sulamistemperatuur

Lihtsustatud lähenemine praimerite sulamistemperatuuri arvutamisele on lisada 2 ºC iga A-T paari ning 4 ºC iga G-C paari kohta (Suggs et al, 1981). Seda nimetatakse ka Wallace'i reegliks. Enamasti kasutatakse täpsemat valemit

$$T_m = \frac{H}{S + R\ln(c/4)} - 273.15°C + 16.6\log_{10}\left[K^+\right]$$

kus H on nukleotiidide hübridiseerumisest tekkinud entalpia muutude summa, S analoogiliselt entroopia muutude summa, R gaasikonstant (8,31 J/mol), c ahelate keskmine molaarne kontsentratsioon. Iseendaga komplementaarse ahela puhul kasutatakse c/4 asemel valemis c. Sellist valemit kasutab oma töös näiteks programm Primer3.

Kogu praimeri sulamistemperatuuri arvutamiseks kasutavad paljud programmid nn. lähimate naabrite meetodit (Breslauer, et al, 1986). See tähendab, et H ja S saadakse kui üksikute naabrite panuste summa.

Praimeripaari korral peab praimerite sulamistemperatuur erinema vähesel määral, et tagada võimalikult ühtlane erisuunaliste praimerite seondumine.

2.2 Kordusjärjestused

Kordusjärjestused (nt. Alu), moodustavad inimese genoomist 50% või üle selle. Et vältida mittespetsiifilist seondumist genoomis korduvatele lõikudele, testivad mitmed programmid (nt. PRIMER) praimeri komplementaarsust genoomsete kordusjärjestustega või välditakse kordusi heksa/oktameeride sagedustabeli kaudu.

2.3 Polü-N traktid

Sarnaselt kordusjärjestustele põhjustab ka ühe nukleotiidi paljukordne järjest esinemine seostumise mittespetsiifilisust. Samuti korduvad di ja trinukleotiidid. Tüüpiliselt esineb matriitsis pikki komplementaarseid kordusi, pikemaid kui praimeris

ning mitmes kohas, mis võimaldab praimeril seonduda paljudesse erinevatesse kohtadesse.

2.4 Praimerite vahelised interaktsioonid

Ühe praimeripaariga (*single-plex*) reaktsioonis tuleb kontrollida praimeri seondumist iseenda ja vastaspraimeriga ning 3' otsa seostumist enda ja vastaspraimeri 3' poole järjestusega, multipleks-reaktsioonis kõiki praimereid omavahel. Iseenda või mõne teise reaktsioonis osaleva praimeriga hübridiseerudes väheneb *template* DNA'ga seostuvate praimerite kontsentratsioon ning võivad tekkida praimerite dimeerid (Kämpke et al, 2000) või sekundaarstruktuurid nt. juuksenõelastruktuur (*hairpin*).

2.5 Produkti pikkus

Lühemad PCR produktid seostuvad väiksema tõenäosusega valede praimeritega. Sobiv PCR produkti pikkus oleneb genoomsete järjestuste puhul asukohast, näiteks mikrosatelliitide puhul on eelistatud kuni 150 aluspaari, heterodupleksi analüüside korral 250-500 aluspaari pikkune produkt (Wallace, unpublished). PCR reaktsioonis on soovitav omada sarnase pikkusega produkte võrdsete koguste tagamiseks ja masina tsükli aegade ühtlustamiseks, multipleks-PCR puhul võimaldab aga erinev produktide pikkus neid kergemini eristada. Üldjuhul jääb sobiv produkti pikkuse vahemik 100 ja 1000 nukleotiidi vahele.

2.6 Praimerite pikkus

Tavaliselt loetakse küllaldaseks praimeri pikkusi alates 18 nukleotiidist. Kindel 18 aluse pikkune oligonukleotiid esineb keskmiselt korra 4^{18} = ~7 x 10^{10} (70 Gb) pikkuses järjestuses, ta ei tohiks seostuda rohkem kui ühte positsiooni enamikes eukarüootsetes genoomides. Samas ei ole genoomi järjestus juhuslik, mis võib nõuda pikemaid praimereid, et vältida mittespetsiifilist seondumist. Lühemad praimerid seostuvad sellise suurusega genoomides suure tõenäosusega mitmesse kohta, kuid neid võib kasutada näiteks juhul, kui matriitsiks (*template*) on plasmiidne DNA.

Praimeri pikkus mõjutab ka sulamistemperatuuri, iga lisatud nukleotiid tõstab seda. Pikemad praimerid nõuavad suuremat AT-sisaldust, et sulamistemperatuur jääks mõistlikkusse vahemikku.

2.7 Praimerite GC nukleotiidide sisaldus

Praimeri nukleotiidide sisaldus praimeri 3’ otsas määrab seostumise spetsiifilisuse DNA järjestusega. On soovitatud kasutada kõrge GC-sisaldusega 3’ otsa (enamasti 50-55%), kuid on näidatud, et kõrge GC-sisaldusega 3’ots võib põhjustada valeseostumisi ja halvendada PCR kvaliteeti (Li et al, 1997). GC sisaldus mõjutab samuti sulamistemperatuuri.

3 Seostumise arvutamise meetodid

PCR modelleerimisel tuleb arvutada praimerite seostumiste arvu ja seostumistugevust teiste praimerite, produktide või genoomiga. Olulisimad on praimer-praimer interaktsioonid. Vahel arvutatakse välja seostumise tugevus absoluutskaalal, mõnikord aga lubatakse lihtsustusi ja loetakse võrdseks A-T ja G-C seondumise tugevus ning tihti arvutatakse seostumise tugevus suhtelisel skaalal, n.ö. skoorina.
Seostumiste leidmisel kasutatakse järgnevaid meetodeid:

3.1 Aukudeta joondus

Aukudeta joonduse (*gapless alignment*) leidmine on kõige lihtsam seostumise arvutamise meetod. Esimese järjestuse positsiooni teise suhtes varieeritakse ning iga positsiooni jaoks loetakse komplementaarsed ja mittekomplementaarsed nukleotiidid, vastavalt millele suurendatakse või vähendatakse skoori. Eraldi väärtused võivad olla teadmata nukleotiidide jaoks. Eeldusel, et tugevam seostumine annab suurema skoori, loetakse lõplikuks seostumise tugevuseks maksimaalne leitud väärtus. Mõnel juhul võib tegeliku seondumise käigus osa nukleotiide paardumata jääda, mida antud algoritm ei arvesta.

3.2 Smith-Waterman

Smith-Watermani algoritm (Smith, Waterman, 1981) on nn. dünaamilise programmeerimise (*dynamic programming*) algoritm. Optimaalseima joonduse leidmiseks arvutatakse „headust“ näitav punktisumma (*score*) *kahe* järjestuse iga võimaliku positsiooni kombinatsiooni kohta ning lõpuks leitakse koostatud tabeli väärtustest maksimum. Maksimumi leidumise kohalt tagasi suundudes (*backtracking*) on võimalik leida optimaalne joondus.

Punktisumma ei ole üheselt määratud, vaid sõltub algoritmile antavatest parameetritest – tüüpiliselt positiivne „preemia“ seondumise (*match*) jaoks ning negatiivne „karistus“ mitteseondumise (*mismatch*) ja augu (*gap*) eest.
Algoritmi töökiirust on võimalik tõsta, kui piirata võimalikku augu pikkust.

3.3 BLAST

BLASTi (*Basic Local Alignment Search Tool*) algoritmi võrreldes Smith-Watermani algoritmiga iseloomustab väiksem tundlikkus (*sensitivity*) ja suurem töökiirus. Kasutades lühikestel sõnadel baseeruvat heuristikat, püüab BLAST ligikaudu realiseerida nn. maksimaalsete segmendipaaride (*maximal segment pairs*) algoritmi. Maksimaalsete segmendipaaride joonduste statistikud on arusaadavad ja seega on võimalik arvutada olulisustõenäosus (Altschul et al, 1997). Ent selline joondus ei saa sisaldada auke, mistõttu ka BLAST eeldab, et insertsioonid ja deletsioonid oluliselt joondust ei mõjuta.
Tüüpiline BLASTi kasutus on antud etalonjärjestusega sarnaste järjestuse otsimist kandidaatjärjestuste seast, näiteks genoomi andmebaasist. BLASTi annaks kasutada praimerite ja produktide vahelise seondumistugevuse arvutamiseks, kui suur täpsus ei ole nõutav.
Praeguseks on arendatud ka täiustatud versioone BLASTist, mis on parema tundlikkuse või kiirusega ning võivad lubada ka auke joonduses. Nimetada võiks näiteks MEGABLASTi.

3.4 Suffiksipuud

Suffiksipuud (*suffix tree*) on üks võimalik efektiivne järjestuste esitamise viis. Nagu nimigi ütleb, sisaldab suffiksipuu sinna lisatava stringi kõiki suffikseid ehk alamstringe, mis algavad suvalisest kohast, ent lõpevad stringi lõpus. Lihtsamal juhul väljuvad igast tipust kaared tema alamtippudesse vastavalt sellele, milline on lisatud stringi või tema suffiksi järgmine täht. Keerulisemal (ja efektiivsemal) juhul hoitakse kaartel viiteid lisatud stringi alamstringi alguspositsioonile ja pikkusele. Samuti võivad tipud sisaldada viiteid teistele puus sisalduvatele tippudele.
Suffiksipuu rakendusi informaatikas ja bioinformaatikas on mitmeid. Üldistatud sufiksipuus on võimalik otsida näiteks pikim ühine alamjada ajahinnanguga O(m), kus

m on üldistatud sufiksipuu suurus. Samuti on üldistatud suffiksipuust võimalik otsida kõikidest sinna lisatud stringidest üheaegselt keerukusega O(n + k), kus n on otsitava sõne pikkus ning k selle sõne esinemiste arv. Suffiksipuudel baseeruvalt on võimalik koostada kogu genoomi joondus, nagu seda teevad programmid MUMmer (Delcher, et al., 2002) ja Multiple Genome Aligner (Höhl, et al., 2002). Samuti on suffiksipuude kasutamine efektiivne mikrokiibile seotud oligonukleotiidsete signatuuride leidmisel geeniekspressiooni uurimisel (Kaderali, Schliep, 2002).

3.5 Räsialgoritmid

Räsialgoritmid (*hashing algorithms*) võimaldavad kiirendada otsinguid andmebaasidest. Räsimine (*hashing*) jaotab järjestuse suhteliselt lühikesteks sõnadeks ning nende põhjal on võimalik koostada otsingut hõlbustav „indeks“. Alamjärjestuste asemel võivad indeksis olla ka nn. sõrmejäljed (*fingerprint, message digest*), mis saadakse spetsiaalset räsifunktsiooni kasutades. Räsimise kombineerimisel teiste meetoditega on võimalik sarnaste järjestuste otsingu keerukus viia lineaarseks.

3.6 Kui pikka ala praimerist arvestada

Püüdes arvutuslikult ennustada praimerite seondumist genoomi, võib piisava täpsuse anda juba praimeri osalise järjestuse kasutamine. On näidatud, et vähemalt 13 nukleotiidi arvestamine praimeri 3' lõpust tagab arvestatava kvaliteedi (Andreson et. al, unpublished). Praimeri osalise ja seega lühema järjestuse kasutamine võimaldab effekti arvutuskiiruses, näiteks suuremõõtmelise genotüpiseerimise korral.

4 PCR multipleks

Tüüpilise ühe praimeripaariga (*single-plex*) PCR reaktsiooni asemel võib samas keskkonnas korraga läbi viia mitme erineva lõigu üheaegse amplifitseerimise. See võimaldab säästa genoomset materjali ning on leidnud kasutust näiteks deletsioonide, mutatsioonide ja polümorfismide analüüsil (Henegariu, et al., 1997). Ühes reaktsioonis amplifitseeritavate lookuste arv on siiski piiratud, kuna vajalik on vältida erinevaid interaktsioone praimerite ja produktide vahel. Kui vajalikku arvu lookusi ei ole võimalik amplifitseerida ühes reaktsioonis, on siiski võimalus lasta arvutil minimiseerida vajalike reaktsioonide hulka. Praimeripaaride jaotamisest (tüüpiliselt võimalikult väikesesse hulka) gruppidesse tulebki järgnevalt juttu.

4.1 Olulised parameetrid grupeerimisel

Olulised grupeerimise parameetrid võib jaotada:

4.1.1 Energia piirväärtused (cutoffs)

Kui oligonukleotiidide seostumisenergia ei ole väga suur, toimub paardumine piisavalt nõrgalt, et mitte alandada liiga palju PCR multipleks-reaktsiooni kvaliteeti. Seostumisenergiat arvutatakse mitmel viisil ning erinevatest oligonukleotiididest või nende lõikudest lähtuvalt. Mõned võimalikud variandid on:

1. Praimer-praimer interaktsioonid, mõlemad otsad
2. Praimer-praimer interaktsioonid, üks ots
3. Praimer-praimer interaktsioonid, suvaline regioon
4. Praimer-produkt interaktsioonid, üks ots
5. Praimer-produkt interaktsioonid, suvaline regioon

4.1.2 Produktide pikkuse erinevuse piirväärtused

Soovitav on omada ühes grupis praimereid, mille produktide pikkuste erinevused ei ole liiga väikesed ega ka mitte liiga suured. Kui reaktsiooni käigus tekib kaks väga sarnase pikkusega produkti, muudab see raskeks nende eristamise näiteks elektroforeesil. Liiga suur produkti pikkuste erinevus nõuab jällegi elektroforeesil suuremat skaalat.

4.1.3 Praimerite sulamistemperatuuride erinevus

Ideaaljuhul peaksid kõik praimeripaarid multipleks-PCR reaktsioonis vastavaid lookusi amplifitseerima sarnase efektiivsusega (Elnifro et. al, 2000). Sulamistemperatuuride erinevus põhjustab ka ebavõrdse amplifitseerimisefektiivsuse, mis nõuab omakord praimerite kontsentratsiooni varieerimist.

4.1.4 Piirangud grupi suurusele

Ühes multipleks-PCR reaktsioonis võib korraga võimendada tüüpiliselt 2-14 (Henegariu, unpublished), mõnel juhul kuni 26 lookust (Lin et. al, 1996). Nii suure arvu lookuste kasutamine õnnestub siiski vaid parimatel juhtudel või spetsiaalvõtteid kasutades (neist räägib ptk. 5), tihti tuleb piirduda 5-10 lookusega (Hacia et al., 1998). Suure arvu lookuste puhul kannatab PCR kvaliteet, kuna siis on tarvis suuremat hulka praimereid, see tähendab aga enamaid kitsendusi keskkonnatingimustele, samuti muutub märgatavaks paljude nõrgemate interaktsioonide mõju. Samas on eesmärk tavaliselt minimiseerida reaktsioonide arvu ning sellest tulevalt maksimeerida lookuste hulka.

4.2 Grupeerimise kvaliteeti mõjutavad tegurid

Amplifikatsiooni spetsiifilisus multipleks-PCR reaktsioonis oleneb ka PCR puhvri ja praimerite kontsentratsioonist ning kasutatavatest temperatuuridest. Puhvrina kasutatakse näiteks KCl-sisaldavat PCR puhvrit (Henegariu, et al, 1997).

5 Alternatiivid praimerite grupeerimisele

Eelnevalt vaadeldud grupeerimise korral kasutatakse praimeritena lühikesi oligonukleotiide, kaks praimerit iga lookuse kohta, ent võimalike interaktsioonide kontrollimine muutub keeruliseks kui neid on palju (Wang et. al, 1998). Erinevad autorid on kirjeldanud teistsuguseid lähenemisi mitmete lookuste üheaegseks amplifitseerimiseks.

5.1 PCR supressioonil baseeruv amplifitseerimine (Broude et al., 2001)

Autorid on arendanud meetodi, mis võimaldab multipleks-PCR-s lookuse kohta tavapärase kahe praimeri asemel kasutada ühte praimerit ning lisaks ühte praimerit kogu reaktsiooni kohta s.t. n lookuse amplifitseerimiseks läheb tarvis n+1 praimerit. Meetod põhineb nn. PCR supressiooni efektil (Lukyanov et al., 1994). Kasutatakse nn. adaptorpraimerit, mis seostub geenifragmentide lõppu ligeeritud n.ö. adaptoriga

(Rosenthal, Jones, 1990). Adaptor on u. 40 aluspaari pikk ning kõrge GC sisaldusega. Pärast iga denaturatsioonisammu formeeruvad iseendaga komplementaarsetest fragmentide lõppudest dupleksid ning juuksenõelastruktuurid (*hairpin structures*). Adapterile vastavad praimerid (A-praimerid) ei replikeeri taolisi DNA fragmente, ent juuksenõelastruktuuri üheahelalises silmuses (*loop*) asuva DNA-ga komplementaarsed praimerid (T-praimerid) küll. Saadud produkti amplifitseerivad nii A- kui T-praimerid. Seega toimub PCR ainult nõutavat sihtjärjestust (*target*) sisaldavates fragmentides. A-praimer aga on kõigile sihtjärjestustele.

Antud meetodit on õnnestunult kasutatud kuni 14 lookusega multipleks-PCR jaoks, eesmärgiga saada analüüse tsüstilist fibroosi põdevate inimeste genotüpiseerimiseks. Tüüpilise multipleks-PCR korral oleks nii suure arvu lookuste kasutamine tihti komplitseeritud (Hacia et al., 1998).

5.2 DOP-PCR (Jordan et. al, 2001)

Üks lähenemine unikaalsete genoomse järjestuse lõikude amplifitseerimiseks on kasutada PCR praimeritena lookustevaheliste kordusjärjestustega (näiteks Alu-järjestus) komplementaarseid järjestusi. Niimoodi saavutatud amplifitseeritavate genoomsete järjestuste hulk on siiski piiratud. Suurema amplifitseeritavate järjestuste hulga annab esialgu kogu genoomi amplifitseerimiseks loodud meetod degeneratiivsete oligonukleotiididpraimeritega PCR, DOP-PCR. DOP-PCR puhul kasutatakse praimereid, kus umbes üks kolmandik oligonukleotiidi keskel asuvatest positsioonidest on degeneratiivsed. Oligonukleotiid, mille 3' otsas on 6 unikaalset nukleotiidi, amplifitseerib efektiivselt suure osa eurkarüootsest genoomist. Seda meetodit on kasutatud muuhulgas inimese ja hiire SNP-de genotüpiseerimiseks. Autorid on näidanud täielikult sekveneeritud genoomide puhul võimalikkust leida optimeeritud hulk DOP-PCR produkte.

5.3 Affymetrix Inc. massgenotüpiseerimise meetod (Matsuzaki et al., 2004)

Affymetrix, Inc. on arendanud välja platvormi ühe praimeriga suuremahuliseks SNP-de leidmiseks (üle 10 000 SNP) (Matsuzaki et. al, 2004). Genoom fragmenteeritakse restriktaaside abil ning seejärel amplifitseeritakse fraktsioonid. Kuna sellega on

oluliselt vähendatud genoomi komplekssust, on võimalik läbi viia alleelispetsiifiline hübridisatsioon maatriksil. Uuritavad SNP-d määratakse põhiliselt lähtuvalt arvuti poolt ennustatud vastavaid SNP-sid sisaldavate fragmentide pikkustest. Restriktsioonilise lõhutamise ning järgneva alleelispetsiifilise hübridisatsiooni ideed on kasutatud varem (Dong et al, 2001; Kennedy et al., 2003), antud juhul optimeerisid autorid mitte-lookusspetsiifilist amplifikatsiooni, genereerimaks restriktsioonifragmendid suurusvahemikus, mis oleks võimalikult reprodutseeritav. Arvuti poolt ennustatud SNP Konsortsiumi kataloogidest pärit SNP-sid sisaldavatele fragmentidele rakendati peale pikkuse teisigi piiranguid ning valiti 11 555 kõrgtihedusega oligonukleotiidmaatriksil aleelispetsiifilise hübridisatsiooni läbiviimiseks sobivaimat järjestust. Genotüüpide ning aleelispetsiifilise hübridisatsiooni intensiivsuste vahelise seose leidmiseks arendati eraldi algoritm (Liu et al., 2003). Genotüpiseerimise efektiivsust ja reprodutseeritavust antud platvormil kontrolliti 307 inimese põhjal 13 etnilisest grupist. Platvormi kogu genoomi haarava skaneerimise näiteks replitseeriti korrektselt ahelduvus kromosoomis 2p kroonilise kandidomükoosi ja/või türoidse häirega.

II Praktiline osa

EESMÄRK

Töö eesmärk oli kirjutada programm praimerite grupeerimiseks, arvestades erinevaid grupeerimist takistavaid interaktsioone. Algselt pidi programm arvestama ainult praimerite seostumistugevusega lokaalse ja globaalse leidmise põhjal, hiljem lisandus täiendavaid kriteeriumeid.

Grupeerimise algne eesmärk oli paigutada kõik praimeripaarid minimaalsesse hulka gruppidesse, hiljem lisandusid piirangud grupi minimaalsele ja maksimaalsele suurusele ning lubati mõningate, halvasti gruppi paigutuvate praimeripaaride välja jätmist.

1 Programmi Multiplx üldine kirjeldus

Programmi Multiplx töötab Unix-keskkonnas käsurežiimis. Programmile antakse parameetritena ette ülesanne ning tööks vajalikud failid. Enamikel parameetritel on vaikeväärtused, mida saab muuta käsurealt, mõnel juhul ka konfiguratsioonifailist.

2 Parameetrite ja käskude kirjeldus

Järgnevalt on toodud programmi viimases versioonis kasutatavad käsud. Programmi põhikasutuseks on mõeldud käsk *gg*, aga teised käsud võivad vajalikud olla näiteks üksikute etappide eraldi teostamiseks, konfiguratsiooni ja sisendfailist lugemise testimiseks.

2.1 Käsud

la Arvutab kahe parameetrina ette antud nukleotiidijärjestuse lokaalse joonduse skoori

```
$ multiplx la ATGCTGT GATCTGA
3
```

ga Arvutab kahe parameetrina ette antud nukleotiidijärjestuse globaalse joonduse skoori

```
$ multiplx ga ATGCTGT GATCTGA
2
```

ea Arvutab kahe parameetrina ette antud nukleotiidijärjestuse lokaalse nn. lõppjoonduse (joondatav ala ei tohi asuda järjestuse keskel)

```
$ multiplx ea ATGCTGT GATCTGA
2
```

ld Näitab kahe parameetrina ette antud nukleotiidijärjestuse lokaalset joondust

```
$ multiplx ld ATGCTGT GATCTGA
CTG
CTG
```

gd Näitab kahe parameetrina ette antud nukleotiidijärjestuse globaalset joondust

```
$ multiplx gd ATGCTGT GATCTGA
CTGT
CTGA
```

ea Näitab kahe parameetrina ette antud nukleotiidijärjestuse nn. lokaalset lõppjoondust

```
$ multiplx ed ATGCTGT GATCTGA
CTGT
CTGA
```

er Näitab infot sisendfaili kohta (multiplx.dat või –i parameetriga etteantav)

```
$ multiplx er
multiplx.dat: 1278 item(s)
```

ei Kuvab valitud numbritega ridadest andmed

```
$ multiplx ei 1 2 3
/.../
```

pt Koostab nõutud skooride tabeli. Esimene parameeter näitab skoori tüüpi, neli järgmist parameetrit on optsionaalsed ja näitavad esimese ja teise indeksi miinimum- ja maksimumväärtust. Skooride tüübid on järgmised:

1 – praimer/praimer lokaalne joondus (PRIMER_SELF_ANY)
2 – praimer/praimer globaalne joondus (PRIMER_SELF_END2)
3 – praimer/produkt lokaalne joondus (PRIMER_PRODUCT_ANY)
4 – praimer/praimer 3’-lõppjoondus (PRIMER_SELF_END1)
5 – primer/produkt 3’-lõppjoondus
6 – produkti pikkuste vahe
>=7 – plugin-ide arvutavad skoorid ja tuleviku rakendused

```
$ multiplx pt 4 1 3 1 3
6       3       4
3       4       3
4       3       3
```

jt Ühendab skooride tabelid, mis on saadud nt. pt käsuga, kokku üheks tabeliks, kasutades antud nivooväärtusi (*cutoff*)

```
$ multiplx pt 1 1 3 1 3 > table.1
$ multiplx pt 2 1 3 1 3 > table.2
$ multiplx pt 3 1 3 1 3 > table.3
$ multiplx jt table.1 5 table.2 2 table.3 15
6       0       1
0       5       0
1       0       8
```

jg Jaotab (eeldatavasti käsu jt tulemusena saadud) antud tabeli põhjal praimeripaarid gruppideks. Teise parameetriga antakse ette maksimaalne skoori väärtus, mille korral praimeripaarid võivad veel ühte gruppi kuuluda

```
$ multiplx jt table.1 5 table.2 2 table.3 9 > table.j
$ multiplx jg table.j 0
Group 1: 1 2
Group 2: 3
```

gg Moodustab ja ühendab skooritabelid (mälus, ajutisi faile ei tekitata) ning grupeerib, andes väljundiks sisendi, kuhu on esimesse tulpa lisatud grupi number. Erinevus jg käsust on veel see, et ette antakse grupi minimaalne ja maksimaalne suurus ning vajadusel ka maksimaalne väljavisatavate praimeripaaride arv

```
$ multiplx gg pt1=5 pt2=2 pt3=9 min=1 max=2 drop=0
1       8_9GH1  367     GGAGACCTGTAGTCAGAGCCAAAAA
/.../
```

2.2 Võtmed

-e n Lubab joonduses maksimaalselt n auku (*gap*), vaikimisi 1

-g x Annab skoori joonduses oleva augu (*gap*) pikenemisel ühe nukleotiidi võrra, vaikimisi 2

-i file Määrab sisendfaili nime, vaikimisi multiplx.dat. - failinimena tähendab standardsisendit

-m x Määrab joondamisel skoori nukleotiidide seostumisele (*match*), vaikimisi 1

-n x Määrab joondamisel skoori nukleotiidide mittesobivusele (*mismatch*), vaikimisi –1

-o file Määrab väljundfailinime, vaikimisi standardväljund (-)

-r n Proovib grupeerimisel kasutada n erinevat järjestust, vaikimisi 0 s.t. juhuslikke järjestusi ei kasutata

-u x Määrab skoori tundmatu nukleotiidi või tundmatute nukleotiidide seostumisele, vaikimisi –0.25

3 Skoori arvutamise meetodid

Hetkel toetab programm järgnevate grupeerimist piiravate skooride arvutamist:

1. kahe praimeri vaheline lokaalne joondus
2. kahe praimeri vaheline globaalne joondus
3. praimeri ja produkti vaheline lokaalne joondus
4. praimeri 3' lõpu joondus teise praimeri suhtes
5. praimeri 3' lõpu joondus produkti suhtes
6. produkti pikkus
7. produktide pikkuse erinevus
8. kasutaja poolt nt. Perli skriptina lisatavad piirangud

3.1 Lokaalne joondus

Lokaalse joonduse leidmiseks kasutatakse Needleman-Wunchi algoritmi. Moodustatakse maatriks, mille mõõtmed on vähemalt ühe võrra suuremad kahe

joondatava järjestuse pikkusest. Edasi hakatakse maatriksit ühes suunas täitma ning iga positsiooni juures leitakse skoor maksimumina

lscore(i, j) = MAX(lscore(i – 1, j – 1), lscore(i – 1, j) + gapweight(1),
lscore(i, j – 1) + gapweight(1))

kus gapweight(1) tähendab karistust 1-nukleotiidi pikkuse augu (*gap*) eest. Kui käsurealt on nõutud pikemate aukude lubamist, vaadatakse vastavalt rohkem eelmisi tabeli elemente läbi.

Lõpuks käiakse tabel läbi ning leitakse maksimaalse skooriga koht, millest tagasi minnes saab kätte kogu joonduse.

3.2 Globaalne joondus

Globaalne joondus leitakse sama protseduuri sees, kuid erinevalt lokaalse joonduse leidmisest lubatakse skooride väärtuseks ka negatiivseid arve ning skoori leides ei otsita maksimumi kogu tabelist, vaid maatriksi alumisest ja paremast äärest. Sellega tagatakse, et järjestused joondatakse täies pikkuses (täpsemalt öeldes joondatakse mõlemast järjestusest vähemalt üks ots).

3.3 Lokaalne „lõppjoondus“ (3'-lõpu joondus)

Selle algoritmi puhul kasutatakse sama maatriksit, mis tavalisel lokaalsel joondusel. Erinevus on see, et maksimumskoori otsitakse ainult maatriksi alumisest ja paremast äärest. Sellega tagatakse, et joondusesse satub vähemalt ühe järjestuse 3’-ots.

4 Grupeerimismetoodika

Hetkel realiseeritud grupeerimisalgoritm kujunes välja mõningate paranduste ja täienduste käigus ning optimeerimise peale ei ole praeguses versioonis just ülearu mõeldud. Algoritm paigutab elemente gruppidesse ükshaaval, järjekord võib olla sama, mis sisendfailis, kuid soovi korral ka juhuslik (grupeerimise efektiivsus oleneb teatud määral sisendelementide järjekorrast, mistõttu paljukordne juhuslik grupeerimine võimaldab selekteerida parima tulemuse andnud variandi). Grupeerimist piiravad argumendid on minimaalne ja maksimaalne grupi suurus ning elementide (praimeripaaride) arv, mis on lubatud grupeerimiset välja jätta, eesmärk on leida neile kriteeriumitele vastavad grupid, samas minimeerides gruppide arvu. Esimene element

paigutatakse esimesse gruppi. Iga uus järjekorrast võetud element püütakse paigutada esimesse gruppi, kuhu ta sobib ning kus ei ole juba maksimaalset hulka elemente. Kui sellist gruppi ei ole, moodustatakse elemendist uus grupp. Kui kõik elemendid on gruppidesse paigutatud, proovitakse midagi ette võtta gruppidega, kus on vähem kui minimaalne hulk elemente. Selle jaoks otsitakse ettejuhtuvaid elemente, mis on mõnes minimaalsest arvust suuremat hulka elemente sisaldavas grupis, ent sobiksid ka mõnesse liiga väikesesse gruppi ning leidmisel tõstetakse element ümber. Kui kõik grupid on piisava suurusega, loetakse saadud tulemus sobivaks variandiks (võrreldes teda erinevate variantide läbivaatuse korral seni leitud sobivaima tulemusega). Kui liiga väikestesse gruppidesse jäänud elementide hulk ei ületa elementide arvu, mis on lubatud grupeerimata jätta, eemaldatakse nimetatud grupid ja loetakse tulemus samuti sobivaks. Vastasel juhul loetakse grupeerimise antud katse ebaõnnestunuks. Skemaatiliselt võiks algoritmi esitada järgmiselt:

```
--- P – sisend - grupeerimata elementide (praimeripaaride) arv
--- s_max – maksimaalne grupi suurus
--- s_min – minimaalne grupi suurus
--- d – elementide arv, mis on lubatud grupeerimisest välja jätta
--- G – väljund – gruppide hulk, kus iga grupp sisaldab elemente
G := {}
iga hulga P elemendi p korral
  lisatud := VÄÄR
  iga hulga G elemendi g korral
    kui sobib_gruppi(p, g) ja [g] < s_max ja lisatud = VÄÄR siis
      g := g ∪ {p}
      P := P \ {p}
      lisatud := TÕENE
    lõpp-kui
  lõpp-iga
  g := {p}
  G := G ∪ {g}
  P := P \ {p}
lõpp-iga
c := VÄÄR --- kas kõik grupid on paraja suurusega
korda kuni c = VÄÄR
```

```
  c := TÕENE
  iga hulga G elemendi g korral
    kui [g] <  s_min siis
      c := VÄÄR
      iga hulga G \ {g} elemendi h korral
        kui [h] >  s_min siis
          iga hulga h elemendi p korral
            kui  sobib_gruppi(p, g) siis
              g := g ∪ {p}
              h := h \ {p}
              katkesta-iga
            lõpp-kui
          lõpp-iga
        lõpp-kui
        kui [g] >  s_min siis
          katkesta-iga
        lõpp-kui
      lõpp-iga
    lõpp-kui
  lõpp-iga
lõpp-korda
n = 0 --- liiga väikestes gruppides olevate praimeripaaride hulk
iga hulga G elemendi g korral
  kui [g] < s_min siis
    n := n + [g]
    G := G \ {g}
  lõpp-kui
lõpp-iga
kui n > d siis --- kui tuli eemaldada lubatust rohkem elemente
  G := {} --- grupeerimine ebaõnnestus
lõpp-kui
```

5 Tulemused (*proof of principle*)

5.1 Programmi töökiirus

Kiiruse testimiseks sai koostatud erineva mahuga sisendandmete komplektid ja mõõdeti programmi töö jaoks kulunud aega Unixi käsuga *time*. Esimesel juhul lasti MULTIPLX-il arvutada erinevate praimeripaaride tabelite jaoks kolm skoori ning praimeripaarid grupeerida. Konkreetne käsk oli järgmine:

```
$ time multiplx gg pt1=8 pt3=12 pt4=4 min=1 max=20
```

Teisel juhul võeti üks sisendtabel ning hinnati aja sõltuvust olenevalt grupeerimise korduste arvust.

```
$ time multiplx gg –r nnn pt1=8 pt3=12 pt4=4 min=1 max=20
```

Saadud tulemustest (toodud lisas, tabelid 1, 2, 3, 4, 5, 6) oli näha, et grupeerimine võtab oluliselt vähem aega kui joondamise skooride arvutamine, samuti on juhuslik gruppide proovimine hädavajalik minimaalse gruppide arvuga lahenduse saavutamiseks (58 praimeripaariga näites saavutati minimaalne gruppide arv 11 alles 10 000 juhusliku grupeerimise järel). Seega tuleks alati lasta programmil proovida grupeerida 10000-100000 või isegi enamal erineval viisil, et saavutada minimaalse gruppide arvuga lahendus.

Programmi tööaja t (sekundites) arvutamiseks sai leitud hinnanguline valem

$$t = C*(p^2+0.005*r*p^{1.5}),$$

kus C on arvuti kiirusest ja sisendandmete olemusest (näiteks produktide pikkused) olenev konstant, p on praimeripaaride arv ning r korduste arv, mida grupeerimise puhul tehakse (-r).

Katsetusteks kasutatud arvutil (Microlink PIII 700 MHz, 2GB RAM, 75 GB SCSI disk) tuli C väärtuseks olenevalt sisendandmetest 0.04 - 0.10.

5.2 Programmi mäluvajadus

Programmi mäluvajadus sõltub kasutatud praimeripaaride arvust ruutvõrdeliselt ligikaudu järgmise valemi kaudu:

$$m = 4 * p^2$$

kus m on kasutatav mälu hulk baitides ning p kasutatavate praimeripaaride arv. Väiksemamahulise sisendi korral annavad tooni ka konkreetse praimeripaariga seotud

andmed (näiteks praimeri ja produkti vastupidises suunas komplementaarne järjestus, nn. *reverse-complement*), grupeerimisega seotud andmed (genereeritud juhuslik praimerite järjestus, seni leitud grupid) ning loomulikult programmi enda kood, kuid peamiselt nõub mälu siiski leitud skooride koondtabel (skoore hoitakse int-tüüpi muutujas, mis on 32-bitisel arhitektuuril 4 baiti, teoreetiliselt võiks kasutada ka 2-baidist muutujat). Üldjuhul on siiski arvutuste juures limiteerivaks aeg, mitte arvuti mälu hulk. Näiteks 1GB RAMiga 2GHz protsessoriga arvuti peal saaks grupeerida 15 000 praimeripaari, kuid see võtaks aega peaaegu terve kuu.

5.3 Programm praktilises kasutuses

Algoritmi efektiivsuse hindamiseks sai võrreldud programmi grupeerimise efektiivsust juhusliku grupeerimisega (Kepp, unpublished). Testimiseks sai leitud praimerid teatud hulga positsioonide jaoks inimese genoomis jaoks ning proovitud neid grupeerida erinevate seostumisskoore piiravate väärtustega gruppideks ning moodustades leitud gruppide suurusega sama suured juhuslike praimeripaaridega grupid. PCR läbiviimisel hinnati valenegatiivsete ning valepositiivsete signaalide osakaalu. Valenegatiivsete signaalide osakaalud on toodud lisas oleval joonisel (Joonis 1). Suuremate gruppide korral suurendati kahe praimeri vahelise globaalse seostumise skoori (primer_primer_end2, PPE2) väärtust, ülejäänud piiravad väärtused olid fikseeritud (praimerite 3' otsadest lähtuv joonduse skoor 8, kahe praimeri vaheline lokaalne joondusel 12, praimeri ja produkti vahelisel lokaalsel joondusel 15. Nagu lisas toodud valenegatiivsete signaalide põhjal saadud graafikult on näha, on programmi poolt pakutud praimeripaaride ühes reaktsioonis kasutamine vajalike lookuste amplifitseerimiseks märgatavalt efektiivsem kui juhuslikult moodustatud gruppide korral. Väga suurte gruppide korral (antud näites 75-liikmelised) oli üllatuslikult juhuslik grupeerimine efektiivsem. Sellise tulemuse põhjused jäid esialgu välja selgitamata. Ometi võib väita, et teatud kasutusväärtus on programmil olemas, valepositiivsete signaalide hulk oli ka 75-liikmeliste gruppide puhul Multiplx tulemusi kasutades väiksem kui juhuslikult komplekteeritud gruppide korral.

Arutelu

Erinevatel autorist olenevatel ja mitteolenevatel põhjustel jäi programmi arendamine katki. Osa planeeritud tööst on realiseeritud teiste inimeste poolt: Windowsi versioon, graafiline kasutajaliides, ulatuslikum objekt-orienteeritud programmeerimise kasutamine arendamisel, tulemuste visualiseerimine, seondumistugevuste arvutamine sõltuvalt reaktsioonitingimustest – DNA kontsentratsioon, monovalentsete katioonide kontsentratsioon, seostumistemperatuur. Võimalik oleks arvestada kahe nukleotiidi võrdlemisel mitte ainult nende vastavust, vaid saadud väärtus enne üldskoorile lisamist korrutada läbi teatud kaaluga, olenevalt praimeri positsioonist. Samuti võib osutuda mõttekaks seostumistugevuse hindamisel kasutada erinevaid väärtusi praimeri või produkti 3' otsas asuva paari nukleotiidi jaoks. Lisaks võib uurida ka praimerite ja produktide GC-sisalduse ning pikkuse mõju grupeerimise efektiivsusele. Kõik need ideed nõuaksid põhjalikumat uurimist, läbimõtlemist ja eksperimenteerimist.

Korduva pooljuhusliku grupeerimise efektiivsus sõltub märgatavalt tsüklite arvust. Rohkem tsükleid annab suure tõenäosusega parema tulemuse, samas teatud tsüklite arvust alates ei ole tulemust enam parandada võimalik. Praeguses versioonis peab vajalikku tsüklite arvu hindama kasutaja, kuid teoreetiliselt võiks seda ka programm teha. Võimalus oleks kasutada näiteks loogikat, et kui on tehtud n tsüklit, n > c ning tsüklite arvu m-kordsel kasvamisel tulemus enam ei parane, loetakse tulemus piisavalt heaks. Konstant c oleks suhteliselt väike arv, mis peaks siiski välistama teatud ebasoodsa juhuse (näiteks olgu m=10 ning 10 esimest grupeerimisest oli juhuslikult esimene kõige parem, samas 100 tsükli korral tuleks tõenäoliselt hoopis parem tulemus). Teine võimalus oleks üritada vajalikku tsüklite arvu hinnata sisendandmetest lähtuvalt, näiteks kui on leitud tingimustele vastav võimalus grupeerimiseks ning kõigis gruppides on elementide arv maksimumile lähedal, võib tulemuse piisavalt heaks lugeda – samas mõnede sisendandmete korral on selline seis saavutamatu.

Kokkuvõte

PCR ehk polümeraasahelreaktsioon on tänapäeval laialt kasutatav meetod tööks genoomse materjaliga. Rohkem kui ühe paari praimeritega PCR läbiviimist, eesmärgiga amplifitseerida mitu lookust, nimetatakse multipleks-PCR-ks. Multipleks-PCR korral halvendavad reaktsiooni kvaliteeti praimerite ja produktide omavahelised ning üksteisega interaktsioonid. Praimeripaaridest sobivate gruppide väljavalimiseks, lähtudes etteantud minimaalsest ja maksimaalsest grupi suurusest, sai koostatud Unixi all töötav arvutiprogramm, mis arvutab praimerite ja produktide omavahelisi seondumistugevusi erinevatel viisidel ning püüab saadud tulemuste põhjal paigutada praimeripaarid minimaalsesse hulka gruppidesse. Programmi võib lubada ka mõned praimeripaarid grupeerimata jätta. Seondumistugevuste arvutamise meetodid globaalne, lokaalse ja 3'-otsast lähtuva joonduse jaoks on tuletatud Needleman-Wunchi algoritmist, lisaks lubab programm arvestada produkti pikkust, produkti pikkuste erinevusi ning kasutada seostumisskooride leidmiseks kasutaja poolt koostatud programme või skripte. Erinevad praimeripaarid lubatakse ühte gruppi panna, kui erinevatel meetoditel saadud seostumistugevuste skoorid ei ületa kasutaja poolt antud piiravaid väärtusi. Grupeerimisel püütakse praimeripaarid algul esimesse võimalikku gruppi panna ning hiljem püütakse liiga väikestesse gruppidesse lisada praimeripaare minimaalsest hulgast rohkem praimeripaare sisaldavatest gruppidest. Kuna tulemus sõltub praimeripaaride järjestusest, on võimalus korrata grupeerimist erinevate juhuslikult järjestatud praimeripaaridega ning leida saadud tulemustest parim. Programmi tööaeg ja mäluvajadus on ligikaudu ruutsõltuvuses praimeripaaride arvust sisendis, kuigi tööaeg oleneb ka näiteks produktide pikkustest, samuti sellest, milliseid meetodeid seostumistugevuste arvutamiseks kasutatakse ja mitu tsüklit kasutatakse grupeerimisel. 2GHz Pentium IV protsessoriga arvuti võib pooletunnise ajaressurssi korral grupeerida orienteeruvalt 200 praimeripaari, kasutades grupeerimisel 1000000 tsüklit. Mäluvajadus ei ole üldjuhul piirav. Programmi on testitud laboris ning tema poolt leitud gruppide põhjal reaktsioonide läbiviimine on praktilistel juhtudel märgatavalt efektiivsem kui juhuslikult komplekteeritud gruppide puhul.

Automated grouping of PCR primers

Meelis Olev

Resume

PCR (polymerase chain reaction) is a widely used method for working with genomic material nowadays. Multiplex-PCR uses more than two pairs of primers and is used to amplify more than one loci. In case of multiplex-PCR must be considered interactions between primers and products. To find the suitable sets of primers that can be used together in reaction, an algorithm and a computer program were developed. This program works under Unix operating system and calculates binding strengths between primers and products in different ways to divide primer pairs to minimum number of groups, considering minimum and maximum group size. It is allowed to exclude an user specified number of primers from groups. Methods for calculating binding strengths for global, local, and 3'-end alignment were derived from Needleman-Wunch algorithm. Additionally, the program allows to consider product lenghts, differences between product lengths and binding scores calculated by an user written external program or script. Different primer pairs can be in one group if different binding scores do not exceed user given cutoff values. When grouping, all primer pairs are tried to include to first possible group at first and later the groups with too few primer pairs are tried to fill from groups with more than minimum number of items. While the result is depending of order of primer pairs, it is possible to repeat grouping with randomly ordered primer pairs and to find the best of the results. Execution time and memory requirement of the program are in about square dependence from number of primer pairs in input, but the execution time is also influenced for example by lenghts of products, methods used for calculating binding scores and number of grouping repetitions. A computer with 2GHz Pentium IV processor may use approximately a half of hour to group about 200 primer pairs, using 1000000 grouping cycles. Memory requirements are not limiting usually. The program is tested in laboratory and it was found that using groups calculated with program is usually significantly more effective than using groups with randomly chosen primer pairs.

Viited

1. Altschul S. F., Madden T. L., Schäffer A. A., Zhang J., Zhang Z., Miller W., Lipman D. J. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acids Research 25(17), 3389-3402, 1997
2. Breslauer K. J., Frank R., Blocker H., Marky L. A. Predicting DNA duplex stability from the base sequence. Proc Natl Acad Sci U.S.A. 83(11), 3746-3750, 1986
3. Broude N. E., Zhang L., Woodward K., Englert D., Cantor C. R. Multiplex allele-specific target amplification based on PCR suppression. Proc Natl Acad Sci USA 98 (1), 206-11, 2001.
4. Brownie J., Shawcross S., Theaker J., Whitcombe D., Ferrie R., Newton C., Little S. The elimination of primer-primer accumulation in PCR. Nucleic Acids Research 25 (16), 3235-3241, 1997
5. Burpo F. J. “A critical review of PCR primer design algorithms and cross-hybridization case study”, Biochemistry 218, 2001
6. Delcher A. L., Kasif S., Fleischmann R. D., Peterson J., White O., Salzberg S. L. Alignment of Whole Genomes. Nucleic Acids Research 30(11), 2478-2483, 2002
7. Dong S., Wang E., Hsie L., Cao Y., Chen X., Gingeras T. R. Flexible use of high-density oligonucleotide arrays for single-nucleotide polymorphism discovery and validation. Genome Research 11:1418-1424, 2001
8. Elnifro E. M. Ashshi, A. M. Cooper R. J., Klapper P. E. Multiplex PCR: Optimization and Application in Diagnostic Virology. Clinical Microbiology Reviews, 559-570, Oct, 2000
9. Glick B., Pasternack J. Molecular Biotechnology: Principles and Applications of Recombinant DNA. ASM Press, Washington, D.C., 1998
10. Haas S., Vingron M., et al. Primer Design for large scale sequencing. Nucleic Acid Research 20(17), 4567-4573, 1998
11. Hacia J. G., Sun B., Hunt N., Edgemon K., Mosbrook D., Robbins C., Fodor S. P., Tagle D. A., Collins F. S. Strategies for mutational analysis of the large multiexon ATM gene using high-density oligonucleotide arrays. Genome

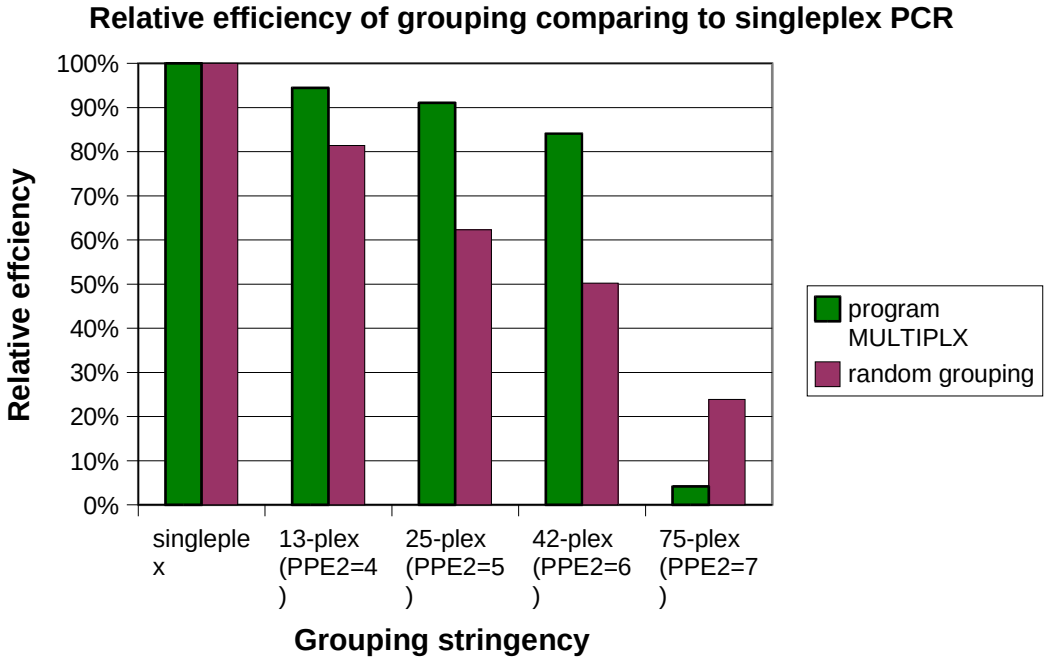
Research 8 (12), 1245-1258, 1998

12. Henegariu O., Heeremaa N. A., Dlouhy S. R., Vance G. H., Vogt P. H. Multiplex PCR: Critical Parameters and Step-by-Step Protocol. Biotechniques 23, 504-511, 1997
13. Höhl M., Kurtz S., Ohlebusch E. Efficient multiple genome alignment. Bioinformatics 18, 312S-320S, 2002
14. Jordan B., Charest A., Dowd J. F., Plumenstiel J. P., Yeh R., Osman A., Housman D. E., Landers J. E. Genome complexity reduction for SNP genotyping analysis. PNAS 99 (5), 2942-2947, 2002
15. Kampke T., Kieninger M., Mecklenbug M. Efficient primer design algorithms. Bioinformatics 17(3): 214-225, 2001
16. Kaderali L, Schliep A. Selecting signature oligonucleotides to identify organisms using DNA arrays. Bioinformatics 18, 1340-1349, 2002
17. Kennedy G. C., Matsuzaki H., Dong S., Liu W. M., Huang J., Liu G., Su X., Cao M., Chen W., Zhang J. et al. Large-scale genotyping of complex DNA. Nat. Biotechnol. 21:1233-1237, 2003
18. Li D., Vijg J. Multiplex co-amplification of 24 retinoblastoma gene exons after pre-amplification by long-distance PCR. Nucleic Acids Research 24 (3), 538-539, 1996
19. Li P., Kupfer K. C., Davies C. J., Burbee D., Evans G. A., Garner H.R. PRIMO: A Primer Design Program That Applies Base Quality Statistics for Automated Large-Scale DNA Sequencing. Genomics 40(3): 476-85, 1997
20. Lin Z., Cui X., Li H. Multiplex Genotype Determination at a Large Number of Gene Loci. Proc Natl Acad Sci USA 93(6): 2582-2587, March 19, 1996
21. Liu W.-m., Di X., Yang G., Matsuzaki H., Huang J., Mei R., Ryder T. B., Webster T. A., Dong S., Liu G, et al. Algorithms for large scale genotyping microarrays. Bioinformatics 19: 2397-2403, 2003
22. Lukyanov, S. A. et al. Bioorganic Chemistry (Russia) 20, 701-704, 1994
23. Matsuzaki H., Loi H., Dong S., Tsai Y., Fang J., Law J., Di X., Liu W., Yang G., Liu G., Huang J., Kennedy G. C., Ryder T. B., Marcus G. A., Walsh P. S., Shiver M. D., Puck J. M., Jones K. W., Mei R. Parallel Genotyping of Over

10,000 SNPs Using a One-Primer Assay on a High-Density Oligonucleotide Array. Genome Research 14:414-425, 2004

24. Michael A. Innis, David H. Gelfand, John J. Sninsky “PCR Applications. Protocols For Functional Genomics” Academic Press, 1999
25. Schuber A. P., Grondin V. J., Klinger K. W. A simplified procedure for developing multiplex PCRs. Genome Research 5 (5), 488-493, 1995
26. Siebert P. D., Chenchik A., Kellogg D. E., Lukyanov K. A., Lukyanov S. A. An improved PCR method for walking in uncloned genomic DNA. Nucleic Acids Research 23 (6), 1087-1088, 1995
27. Smith T. F., Waterman M. S. Identification of common molecular subsequences. Journal of Molecular Biology 147 (1), 195-197, 1981
28. Suggs S., et al. ICN-UCLA Symposia on Developmental Biology Using Purified Genes. Academic Press Inc., New York, NY, Vol 23, pp683-693., 1981
29. Sugimoto N., Nakano S., Yoneyama M., Honda K. Improved thermodynamic parameters and helix initiation factor to predict stability of DNA duplexes. Nucleic Acids Research. 24, 4501-4505, 1996
30. Wang, D. G., Fan J. B., Siao C. J., Berno A., Young P., Sapolsky R., Ghandour G., Perkins N., Winchester E., Spencer J., et al. Science 280, 1077-1082, 1998

Lisad



Joonis 1. Programmi multiplx poolt genereeritud gruppide efektiivsus võrreldes juhusliku grupeerimisega (Kepp, unpublished). Kriteeriumina on kasutatud valenegatiivsete vastuste hulka.

Praimeripaare sisendfailis	Time käsu tulemus	Arvutus-aeg (s)
3	0.640u 0.010s 0:00.65	0.64
11	13.060u 0.000s 0:13.05	13.06
58	135.400u 0.370s 2:15.89	135.40
99	1002.580u 0.010s 16:43.34	1002.58
298	3538.750u 0.020s 58:59.80	3538.75
952	91672.790u 4.770s 25:28:02.94	91672.79

Tabel 1. Arvutusaja (koos grupeerimisega) sõltumine praimeripaaride arvust sisendis.

Korduste arv	Time käsu tulemus	Arvutus-aeg (s)	Grupeerimis-aeg (s) ~	Gruppe
0*	134.900u 0.520s 2:15.41	134.90	0.00	14
1	135.360u 0.000s 2:15.35	135.36	0.46	13
10	135.370u 0.020s 2:15.39	135.37	0.47	13
100	135.350u 0.010s 2:15.36	135.35	0.45	12
1000	135.640u 0.010s 2:15.65	135.64	0.74	12
10000	137.640u 0.000s 2:17.64	137.64	2.74	11
50000	146.730u 0.010s 2:26.73	146.73	11.83	11
100000	158.340u 0.000s 2:38.35	158.34	23.44	11

Tabel 2. Grupeerimisaja sõltuvus tsüklite arvust, kasutades 58 praimeripaariga sisendit.

** korduste arv 0 tähendab siiski ühte grupeerimist, aga sellisel juhul ei ole gruppidesse paigutamise järjekord juhuslik. Kuna mittejuhusliku ühekordse grupeerimise aeg on piisavalt väike, on ta võetud lähtepunktiks grupeerimisaja mõõtmisel.*

Korduste arv	Time käsu tulemus	Arvutus-aeg (s)	Grupeerimis-aeg (s) ~
0*	12.970u 0.010s 0:12.97	12.97	0.00
1	12.970u 0.000s 0:12.97	12.97	0.00
10	13.050u 0.000s 0:13.04	13.05	0.08
100	13.010u 0.010s 0:13.02	13.01	0.04
1000	13.060u 0.010s 0:13.06	13.06	0.09
10000	13.300u 0.000s 0:13.30	13.30	0.33
50000	14.210u 0.020s 0:14.23	14.21	1.24
100000	15.550u 0.010s 0:15.55	15.55	2.58

Tabel 3. Grupeerimisaja sõltuvus tsüklite arvust, 11 praimeripaari.

Korduste arv	Time käsu tulemus	Arvutus-aeg (s)	Grupeerimis-aeg (s) ~
0*	0.640u 0.010s 0:00.64	0.64	0.00
1	0.640u 0.000s 0:00.64	0.64	0.00
10	0.650u 0.000s 0:00.65	0.65	0.01
100	0.640u 0.010s 0:00.65	0.64	0.00
1000	0.660u 0.000s 0:00.65	0.66	0.02
10000	0.710u 0.000s 0:00.70	0.71	0.07
50000	0.930u 0.000s 0:00.92	0.93	0.29
100000	1.210u 0.000s 0:01.20	1.21	0.57

Tabel 4. Grupeerimisaja sõltuvus tsüklite arvust, 3 praimeripaari.

Korduste arv	Time käsu tulemus	Arvutus-aeg (s)	Grupeerimis-aeg (s) ~
0*	1002.580u 0.010s 16:43.34	1002.58	0.00
100000	1060.780u 0.040s 17:40.80	1060.78	58.20

Tabel 5. Grupeerimisaja sõltuvus tsüklite arvust, 99 praimeripaari.

Korduste arv	Time käsu tulemus	Arvutus-aeg (s)	Grupeerimis-aeg (s) ~
0*	3538.750u 0.020s 58:59.80	3538.75	0.00
100000	3917.820u 0.620s 1:05:18.43	3917.82	379.07

Tabel 6. Grupeerimisaja sõltuvus tsüklite arvust, 298 praimeripaari.