

0101001110110000010101000101001001100100110101001101111010100
ACTGCTACCTTCTACTTTAGGGGCGCGTAGGGCGTATCTCGTC

```
($snp_id, $seq_file) = @_;
```

```
my @temp_id = split (/s+/, $snp_id);
```

```
my $dsn = "DBI:mysql:lite_28:localhost";
```

```
my $user_name = "root";
```

```
my $password = "root";
```

```
my ($dbh, $sth); # $dbh means database handle, $sth who knows?
```

```
open OUT, ">$seq_file" or (print "Error: Couldn't open to _primer3 file!\n" and die);
```

```
# database connection, query etc. stuff
```

```
$dbh = DBI->connect ($dsn, $user_name, $password) or $output[0] = "$snp_id Fail
```

```
my $db = -1;
```

```
my (@db_id, @db_chr, @db_start, @db_allele) = ();
```

```
for $x (@temp_id){
```

```
    $sth1 = $dbh->prepare (qq{
```

```
        SELECT chr_name, snp_chrom_start, allele
```

```
ATGCTGAGCGGGCCTGGCTCTAGCTTGAGTCGGATCGTACGC
```

**Kiire ja tundlik dot-matrix
meetod genoomse järjestuse
analüüsiks**

Reidar Andreson

17.03.2004.

Dot-matrix

- Üks esimesi meetodeid üldse järjestuste analüüsis
- Kasutatakse:
 - kahe järjestuse (DNA või valk) sarnasuste leidmiseks graafilisel teel
 - insertioonide/deletsioonide leidmiseks
 - korduste leidmiseks
 - sekundaarstruktuuride leidmiseks

10100111011000001010100010100100110101001101111010100
ACTGCTACCTTCTACTTTAGGGGCGCGTAGGGCGTATCTCGTC

```
($snp_id, $seq_file) = @_;
```

Dot-matrix

```
my @temp_id = enlit (/As+/ $snp_id);
```

```
my $dsn = "  
my $user_n  
my $passw  
my ($dbh, $
```

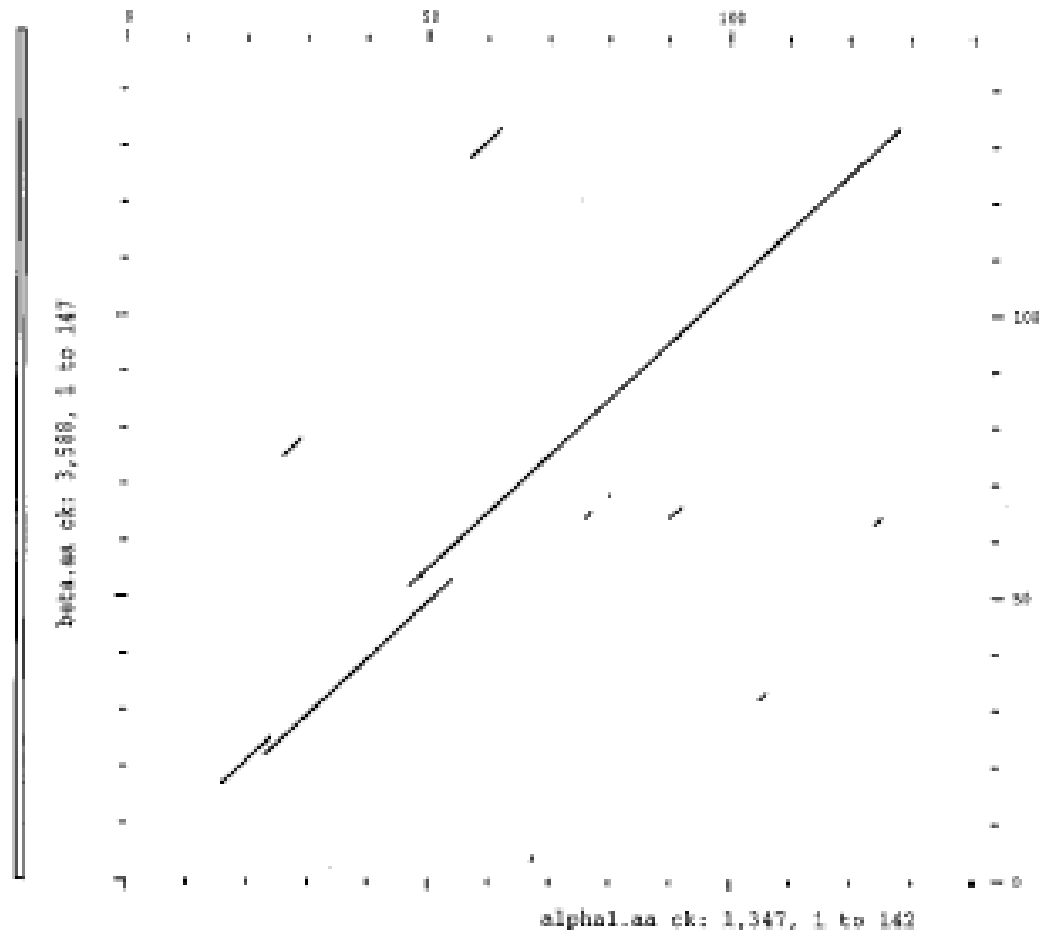
```
open OUT,
```

```
# database  
$dbh = DBI-
```

```
my $db = -1  
my (@db_id
```

```
for $x (@te  
$sth1 =
```

DOTPLOT of: alpha1.pat Density: 165.92 March 8, 2003 16:13
compare window: 50 stringency: 11 Points: 143



ows?

```
); file!\n" and die);
```

```
it[0] = "$snp_id Fail
```

```
SELECT chr_name, snp_chrom_start, allele
```

ATGCTGAGCGGGCCTGGCTCTAGCTTGAGTCGGATCGTACGC

1010010111010101010100111101010001001010110001101001

Dot-matrix programmid

- DOTTER (Sonnhammer and Durbin, 1995)
 - Efektiivne lühikeste järjestuste korral (<1 Mb)
 - Liigub nukleotiidhaaval
 - Kasutab vähe mälu ja kettaruumi
 - Kasutab “liikuvat akent” vältimaks juhuslikke dot’ide kuvamist (vähendab müra osakaalu)
- DOTTUP (Rice *et al.*, 2000)
 - Kasutab sõna-otsingu põhimõtet järjestuste sarnasuse leidmisel
 - Ainult 100% identsed match’id

LBDOT – meetod 1

- Luuakse tabel kõikvõimalikest sõnade (k-tuple) kombinatsioonidest
- p^k ($p=4$ DNA, $p=20$ valk)
- Iga võimalik sõna konverteeritakse arvuks 1 ja p^k vahel
- C sisaldab viimaseid viiteid sõnade positsioonidele järjestuses
- D_{initial} sisaldab viidete positsioone, mis sõna antud positsioonis esineb
- D_{final} sisaldab viidet positsioonile, kus viimati antud sõna esines

LBDOT – meetod 1

my	A	G	C	T	C	G	A	T	C	G	A	G	T	C	T	C	G	A	G	T	A	G
Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
D: initial set	3	10	8	14	7	9	4	14	7	9	3	12	14	8	14	7	9	3	12	13	3	
D: final value	0	0	0	0	0	0	0	4	5	6	1	0	8	3	13	9	10	11	12	0	18	
C:	0	0	21	7	0	0	16	14	17	2	0	19	20	15	0	0						
# k-tuple	AA	AC	AG	AT	CA	CC	CG	CT	GA	GC	GG	GT	TA	TC	TG	TT						

LBDOT – meetod 1

- Toimub järjestuste omavaheline võrdlus sõnade kaupa
- match = 0, mismatch = -1
- w akna pikkus (suurem kui sõnapikkus k)
- s piirväärtus (threshold) – negatiivne väärtus
- Sõnu võrreldakse seni kui score langeb alla s väärtuse
- Kui matchi pikkus $> w$, on tegu “tõeliselt” sarnase järjestuse segmendiga

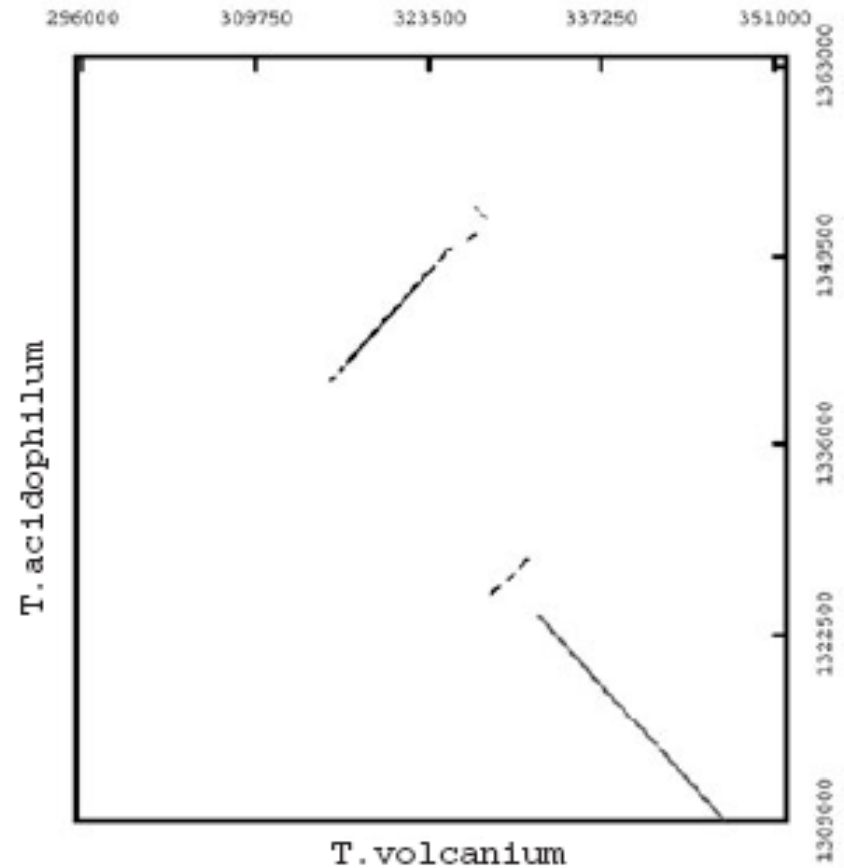
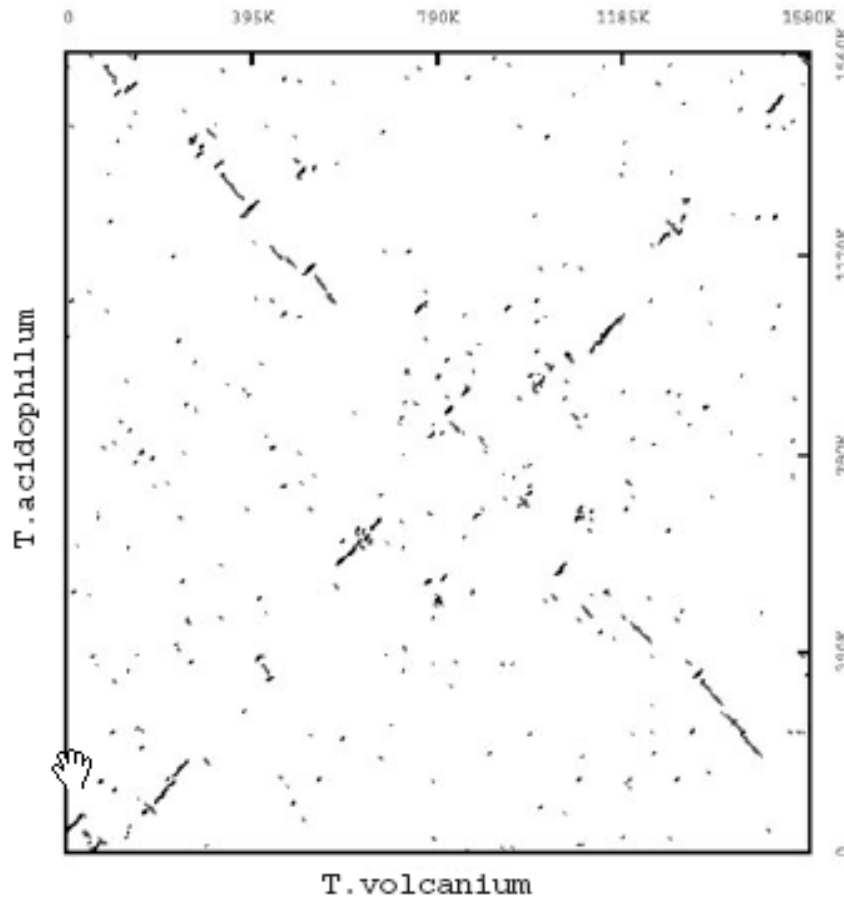
LBDOT – meetod 2

- Meetod 1 ei luba mismatch'e tabeli algsel koostamisel
- Mõlemad järjestused konverteeritakse aminohapeteks
 - 3 lugemisraami (+3 antisense ahelalt)
 - Luuakse tabel nagu meetod 1 puhul
 - Leitakse “tõeliselt” sarnased segmendid

LBDOT

- Mäluvajadus – 5-10 MB 1 Mb järjestuse kohta
- Sõnapikkused:
 - Meetod 1: 6-14 bp
 - Meetod 2: 10-21 bp
- Meetod 2 on tundlikum prokarüootide järjestuste võrdlemisel

LBDOT (meetod 2)

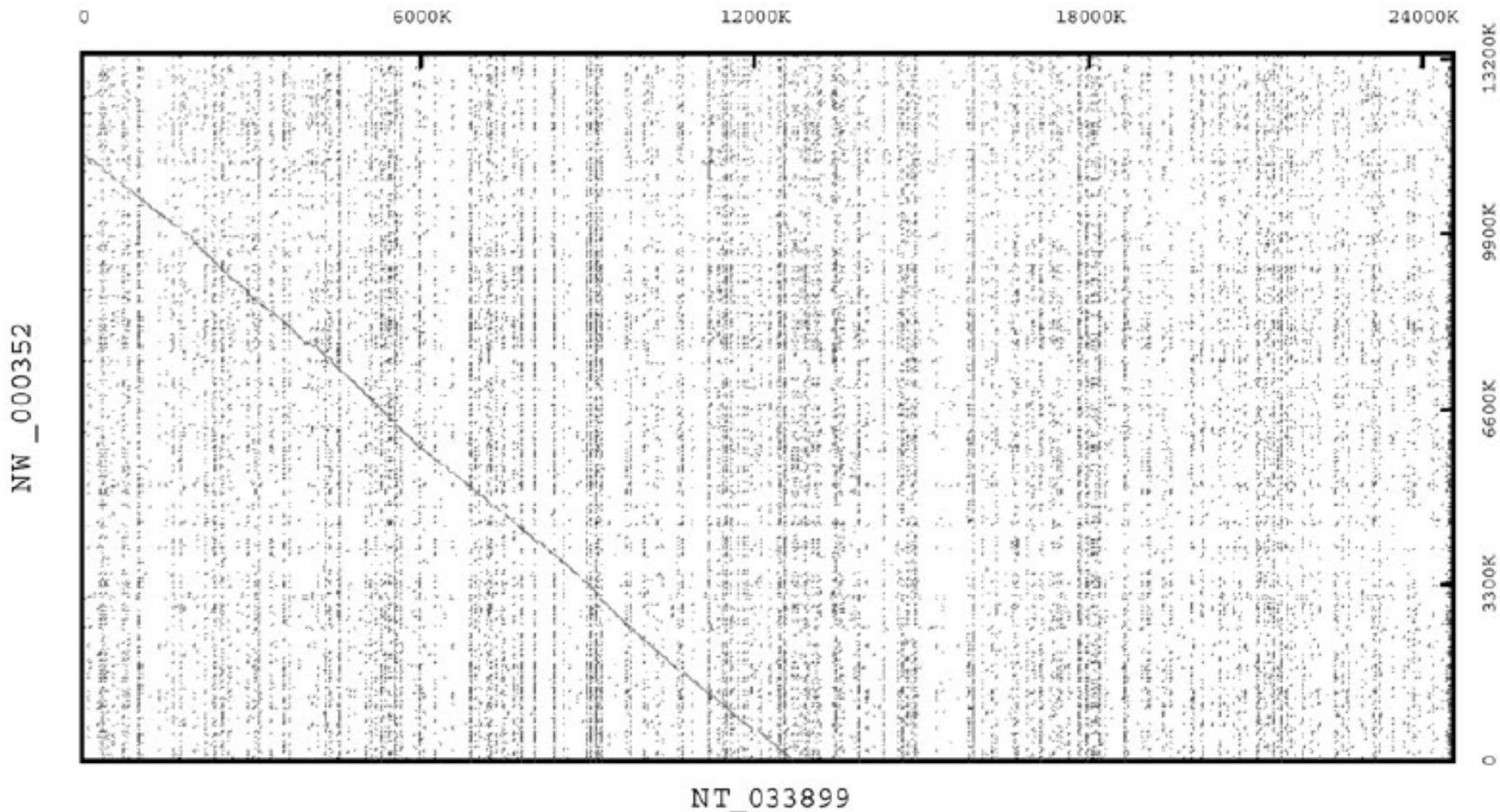


SELECT chr_name, snp_chrom, start, allele
K-tuple = 5, w = 27, aeg = 15,6 sek

10100111011000001010100010100100110101001101111010100
ACTGCTACCTTCTACTTTAGGGGCGCGTAGGGCGTATCTCGTC

(\$snp_id, \$seq_file) = @

LBDOT (method 1)



K-tuple = 12, w = 69, s = -10, aeg = 8 min

ATGCTGAGCGCGCTGCTAGCTTCGAGTCGGATCGTACGC
01001011101010101010101010011110101001001010110001101001

Kiiruste võrdlus

- **DOTTER** vs **LBDOT (m1 ja m2)**

– 700 MHz protsessor

1,5 Mb: 7 päeva vs 2.9s, 4.5s

- **DOTTUP** vs **LBDOT (m1 ja m2)**

– 700 MHz G4 Mac (384 MB RAM)

1,5 Mb: 6300s vs 0.3s, 3s

LBDOT kiirus

- Eukariootide kordused aeglustavad algoritme
- Lahendus: maksimum väärtus sõnade esinemise kohta järjestuses
 - C ja D vastavad positsioonid nullitakse
 - Ei kaota märkimisväärset tundlikkuses?

LBDOT kiirus

- Kui eesmärk pole leida kordusi...
- välja jätta sellised korduvad motiivid nagu:
AAAAAAAAAAAAAAAA ja TTTTTTTTTTTTTT
(esineb kromosoomis 11 vastavalt 15561 ja 16064 korda); meetod on ~25% kiirem
- Filtreeriti välja 16 levinud kordusmotiivi –
~50% kiirem

1. Huang Y, Zhang L.

Rapid and sensitive dot-matrix methods for genome analysis.

Bioinformatics. 2004 Mar 1;20(4):460-6. Epub 2004 Jan 22.

2. Sonnhammer EL, Durbin R.

A dot-matrix program with dynamic threshold control suited for genomic DNA and protein sequence analysis.

Gene. 1995 Dec 29;167(1-2):GC1-10.