Tools for constructing genome contigs: an introduction to *phred/phrap/consed*

Jaroslaw Pillardy

Computational Biology Service Unit
Cornell Theory Center

Phred/phrap/consed is a set of programs for complete assembly and finishing of a sequencing project, starting from binary chromatogram files as input.

- *Phred* is a base calling program interpreting binary chromatogram files.
- *Phrap* is an assembler
- *Consed* (and its automated version *autofinish*) is used for visualization of assembly and for finishing

phred and *phrap* are best used together from *phredPhrap* Perl script.

Cross_match/*SWAT*

In order to perform assembling *phrap* needs a program for aligning sequences. Such a program is also necessary for masking vector/primer contaminations, tagging repeats etc.

Program *cross_match* is used for sequence alignment within the package. It is a version of *SWAT* program, an efficient implementation of the Smith-Waterman algorithm for comparing any two sets of (long or short) DNA sequences. This is slower implementation, but more sensitive than BLASTN.

Other publicly available packages

- ***GAP4*** (Genome Assembly Program). The program contains all the tools that would be expected from an assembly program plus many unique features and a quite nice interface.

- ***CAP3***. Another good assembler. It is known to work well with ESTs.

We are planning to have both programs available on our CBSU servers this fall.

Where is *phred/phrap/consed*

Phred/phrap/consed is available through the Internet free, but requires signing of several license agreements.

There is a licensed version of the package installed on the CBSU Linux server available to our collaborators.

We are working on installing Windows version of the package and expect it to be ready this fall.

Phred

- *Phred* reads DNA sequencer trace data, calls bases, assigns quality values to the bases, and writes the base calls and quality values to output files.

- *Phred* can read trace data from SCF, ABI model 373 and 377 DNA sequencer chromatogram, and MegaBACE ESD chromatograms files.

- After calling bases, *phred* writes the sequences to files in either FASTA format or the SCF format. Quality values for the bases are written to FASTA format files or PHD files, which can be used by the *phrap* sequence assembly program in order to increase the accuracy of the assembled sequence.

Phred: algorithm

- *Phred* uses simple Fourier methods to examine the four base traces in the region surrounding each point in the data set in order to predict a series of evenly spaced predicted locations.

- Next *phred* examines each trace to find the centers of the observed peaks and the areas of these peaks relative to their neighbors. The peaks are detected independently along each of the four traces so many peaks overlap. A dynamic programming algorithm is used to match the observed peaks detected in the second step with the predicted peak locations found in the first step.

- *Phred* evaluates the trace surrounding each called base to quantify the trace quality. The quality value is related to the base call error probability by the formula $QV = - 10 \log_{10}(P_e)$.

- For dye primer data, *phred* identifies loop/stem sequence motifs that tend to result in CC and GG merged peak compressions. It reduces the quality values of potential merged peaks and splits those peaks that have certain trace characteristics indicative of merged CC and GG peaks. In addition, the chemistry and dye information are passed to *phrap*.

Phrap

- Allows use of entire read (not just trimmed high quality part).
- Uses a combination of user-supplied and internally computed data quality information to improve accuracy of assembly in the presence of repeats.
- Constructs contig sequence as a mosaic of the highest quality parts of reads (rather than a consensus).
- Provides extensive information about assembly (including quality values for contig sequence) to assist trouble-shooting.
- Able to handle very large datasets.

Phrap: algorithm

1. Read in sequence & quality data, trim off any near-homopolymer runs at ends of reads, construct read complements.
2. Find pairs of reads with matching words. Eliminate exact duplicate reads. Do *swat*/*cross_match* comparisons of pairs of reads which have matching words, compute (complexity-adjusted) *swat*/*cross_match* score.
3. Find probable vector matches and mark so they aren't used in assembly.
4. Find near duplicate reads.
5. Find reads with self-matches.
6. Find matching read pairs that are "node-rejected" i.e. do not have "solid" matching segments.

Phrap: algorithm

7. Use pairwise matches to identify confirmed parts of reads; use these to compute revised quality values.
8. Compute scores for each match (based on qualities of discrepant and matching bases).

   (Iterate above two steps).
9. Find best alignment for each matching pair of reads that have more than one significant alignment in a given region (highest scores among several overlapping).
10. Identify probable chimeric and deletion reads (the latter are withheld from assembly).
11. Construct contig layouts, using consistent pairwise matches in decreasing score order (greedy algorithm). Consistency of layout is checked at pairwise comparison level.
12. Construct contig sequence as a mosaic of the highest quality parts of the reads.
13. Align reads to contig; tabulate inconsistencies (read / contig discrepancies) & possible sites of misassembly. Adjust scores of contig sequence.

Consed / autofinish

- *Consed* is graphical program for finishing tightly integrated with Phrap.
- Editing time reduced by the program's ability to pin-point problem areas.
- Uses a more efficient method of editing.
- Editing is guided by error probabilities.
- *Consed* is able to pick primers very successfully.

Autofinish is part of the *consed* package. It automatically chooses finishing reads in order to finish a project.

- Figure out how contigs are ordered and oriented
- Close gaps
- Improve the error rate
- Cover every base by reads from at least 2 different subclones

Autofinish

BAC or genome	size base pairs	*** shotgun *** # of reads	# of contigs	error rate	***finishing*** # of reads	% auto	final error rate
djs40	185652	5797	3	26.68	86	100	0.00
djs45	195981	4872	1	0.95	11	100	0.11
djs58	179757	3956	2	5.17	72	100	0.01
djs14	150242	2731	4	42.01	91	100	0.23
djs301	188094	3092	10	77.99	224	98	0.03
djs124	153339	2387	9	222.98	149	97	0.03
djs77	169347	2808	2	4.25	113	95	0.01
djs94	178635	2760	7	87.14	127	87	0.03
djs146	156940	2711	5	43.61	141	86	0.01
djs228	177286	3691	5	54.78	128	81	0.02
djs119	175212	2970	3	16.49	82	80	0.01
PA01	6264410	95104	54	200	1637	73	0.38
djs104	155401	3172	4	8.75	140	64	0.02
rg391h	197346	3104	5	30.56	390	61	0.10
djs201	167237	2989	10	168.68	267	60	0.04

Setting up an assembly project

1. A new directory (e.g. example1) should be created for data processing and storing
2. Three subdirectories for data processing have to be created there:
   a) chromat_dir – for input chromatogram files
   b) edit_dir – for output files
   c) phd_dir – for intermediate files
3. Local copy of phredPhrap should be placed in example1/ and customized
4. File determineReadTypes.perl has to be customized in order to properly recognize read type and direction.
5. Dye library should be inspected, and ,if necessary, updated.
6. Files containing vector, repeats and primer information should be updated/customized.

Data required by phrap for assembling:

(i) sequence and quality data [*]

(ii) name of the subclone (or other template) from which the read is derived (this is used, for example, in checking for chimeric subclones) [N]

(iii) orientation of the read (forward or reverse) within the subclone, in cases where data is acquired from each end of a subclone insert [N]

(iv) chemistry used to generate the read (which influences phrap's decisions as to how to treat discrepancies between potentially overlapping reads) [*]

[*] extracted from binary chromatogram file by phred

[N] inferred from chromatogram file name

"St. Louis" read naming convention:

1. The portion of the read name up to the first '.', if any, is the name of the subclone from which the read is derived

2. The orientation of the read within the subclone, and the chemistry, are indicated by the first letter following the '.'

"**s**" forward direction read on single stranded (SS) template, dye primer chemistry
"**f**" forward read on double stranded (DS) template, dye primer chemistry
"**r**" DS reverse read, dye primer chemistry
"**x**" SS forward read, standard dye terminator chemistry
"**z**" DS forward read, standard dye terminator chemistry
"**y**" DS reverse read, standard dye terminator chemistry
"**i**" SS forward read, big dye terminator chemistry
"**b**" DS forward read, big dye terminator chemistry
"**g**" DS reverse read, big dye terminator chemistry

"**t**" for T7 (cDNAs)
"**p**" for SP6 (cDNAs)
"**e**" for T3 (cDNAs)
"**d**" for special
"**c**" consensus pieces
"**a**" assembly pieces

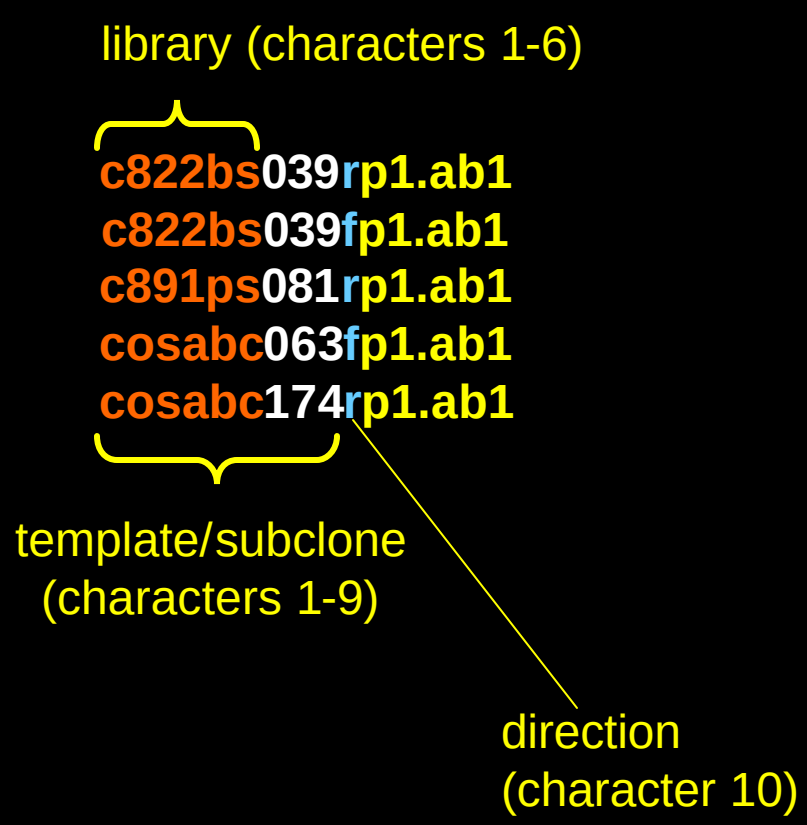
Example:

BAC112_a11b3.x3_pg

from the subclone BAC112_a11b3.

forward read with standard dye terminator chemistry

Example 1: Names



File determineReadTypes.perl has to be customized in order to properly recognize read type and direction.

Dye library: file /usr/local/genome/lib/phredpar.dat

```
##################################################################
#                                                                #
# phredpar.dat - phred parameter file: 980806                    #
#                                                                #
#   known chemistries: primer, terminator, unknown               #
#   known dyes       : rhodamine, d-rhodamine, big-dye           #
#                      energy-transfer, bodipy, unknown          #
#   known machines   : ABI_373_377, MolDyn_MegaBACE,             #
#                      ABI_3700, LI-COR_4000                     #
#                                                                #
# Notes:                                                         #
#       (1) enclose the `dye primer' name in double quotes       #
#           and include spaces in the names.                     #
#       (2) leave one or more spaces between the `dye primer'    #
#           and chemistry names, between the chemistry and       #
#           dye names, and between the dye and machine names.    #
#       (3) add entries between the `begin chem_list' and        #
#           `end chem_list' lines.                               #
#                                                                #
##################################################################
#
begin chem_list
"DP6%25Ac{-21M13}"              primer          rhodamine               ABI_373_377
"DP6%Ac{-21M13}"                primer          rhodamine               ABI_373_377
"DP6%25Ac{M13Rev}"              primer          rhodamine               ABI_373_377
"DP6%Ac{M13Rev}"                primer          rhodamine               ABI_373_377
"DyePrimer{-21m13}"             primer          rhodamine               ABI_373_377
"DyePrimer{KS}"                 primer          rhodamine               ABI_373_377
"DyePrimer{M13RP1}"             primer          rhodamine               ABI_373_377
"DyePrimer{SK}"                 primer          rhodamine               ABI_373_377
"DyePrimer{SP6}"                primer          rhodamine               ABI_373_377
"DyePrimer{T3}"                 primer          rhodamine               ABI_373_377
```

Sequence library: /usr/local/genome/lib/screenLibs

Files:

vector.seq

important for phrap, can be placed in local directory

primerCloneScreen.seq and primerSubcloneScreen.seq

important for consed/autofinish, location controlled by environmental variable CONSED_HOME

repeats.fasta

tagging repeats for consed/autofinish

phredPhrap

1. Run *phred* on all new reads. Binary chromatogram files from chromat_dir are translated into ascii files in phd_dir. This is base calling stage.
2. Run *determineReadTypes.perl*. Read types and orientations are written into phd files.
3. Run *phd2fasta* to create fasta files for all reads and accompanying quality files.
4. Run *crossmatch* to screen reads for vector sequences. Vector sequences are converted to X in fasta files.
5. Run *phrap*. This is assembly stage.
6. Prepare output for *consed.* Consensus tags are transferred and repeats are tagged.

phredPhrap modified for ESTs

1. Run *phred* on all new reads. Binary chromatogram files from chromat_dir are translated into ascii files in phd_dir. This is base calling stage.

1a. Run *qualtrim*. Applies stringent quality requirement for eliminating low quality reads (200bp continuous block with each bp quality at least 20, no more than 10 bp with lower quality are allowed).

1b. Run *trim*. Trims low-quality ends from reads (each ending 25bp block must contain no more than 5bp of low quality).

2. Run *determineReadTypes.perl*. Read types and orientations are written into phd files.

3. Run *phd2fasta* to create fasta files for all reads and accompanying quality files.

phredPhrap modified for ESTs

4. Run *crossmatch*
   a) to screen reads for viral/bacteria DNA contamination.
   b) to screen reads for vector sequences.
   c) to screen for primer contamination.

   Matches are removed from sequences, too short sequences are eliminated
4. Run *phrap*. This is assembly stage.
5. Format output. Contigs and singlets are merged into a single fasta file and sequence names are updated in order to keep track of templates.