R2R

From Reads to Results

Manual for detection of mutations from deep short read sequencing using the R2R scripts

version 0.95

Download:

http://milne.ruc.dk/R2R/

Contact:

olesk@ruc.dk

Contents

The impatient reader

The impatient reader should immediately inspect the appendix E: “A quick start tutorial”.

What is R2R?

The “From Reads to Results” package is demonstration that short read sequences can be aligned to a reference sequence and SNP’s, insertions and deletions can be called and visualized in a user friendly way without requiring only a standard web browser – and it can all be performed on a lap-top size computer faster than sequence data can be generated so far. Sequence quality is utilized for filtering reads and for evaluating called mutations. Algorithm-wise R2R demonstrates the use of sorting as an alternative to currently used approaches such as spaced seeds or Burrows Wheeler transform. R2R is greedy for disk-space, but modest in memory usage.

Since it is a “demonstration” it does not come with an abundance of utilities and features expected from a full program. Instead it comes – almost - out of a box ready to find mutations with a higher sensitivity than any other programs I know of, including MAQ, BWA, Mosaic and SHRiMP.

The analysis ends with a list of “interesting positions to study further” and corresponding panels with alignments of “alignable” reads with reference sequence together with supplementary information such as annotated translations or RNA genes. This comes in hyperlinked html format files. Figure 1 shows an example of a panel. Here is called a point mutation and an insertion of one extra G in a CAGGGGAA sequence.

R2R has so far proven capable of analyzing bacterial and smaller eukaryotes, but in principle it should scale to any re-sequencing project. Beginning with v. 0.950 R2R analyze paired end / mating pairs and similar data. Orphan reads, i.e. unmapped reads are returned in FASTQ format for further analysis against other reference sequences, for assembly or for testing with alternative programs.

R2R has the potential to analyze transcription/ChIP data, and it has the potential to be highly parallelized though all this has yet to be implemented.

New in R2R v.9.50

R2R now handles paired reads (paired end, mate pairs etc.)

Information on possible changes in the amino acid sequence caused by called mutations.

Experiments with use of quality to filter InDel calls.

A number of other changes in the variables to be set.

A

Panel_No	P_Pos	Mut/InDel	#Mut	#Mut+	#Mut-	#Match	ID_qual	Repeat	SNP?	Suggested mutation(s)		Annotation(s)
29	730	InDel:	6	6	0	6	1.18					carA (carbamoyl
299	443	Mutation:	27	14	13	0				c>t (1.00)	G>D;	yagR (predicted
369	399	InDel:	6	0	6	5	0.82					mhpA (3-(3-hydr
487	626	Mutation:	4	0	4	4				a>c (1.00)	T>P;	kefA (fused con
537	590	InDel:	10	10	0	4	1.14					ybbW (predicted
541	887	InDel:	6	6	0	4	1.02					glxK (glycerate
547	694	Mutation:	27	15	12	0				a>g (1.00)		ylbE (Pseudogen
547	832	InDel:	24	15	9	0	0.97					ylbE (Pseudogen
547	836	InDel:	18	13	5	0	1.02					ylbE (Pseudogen

B

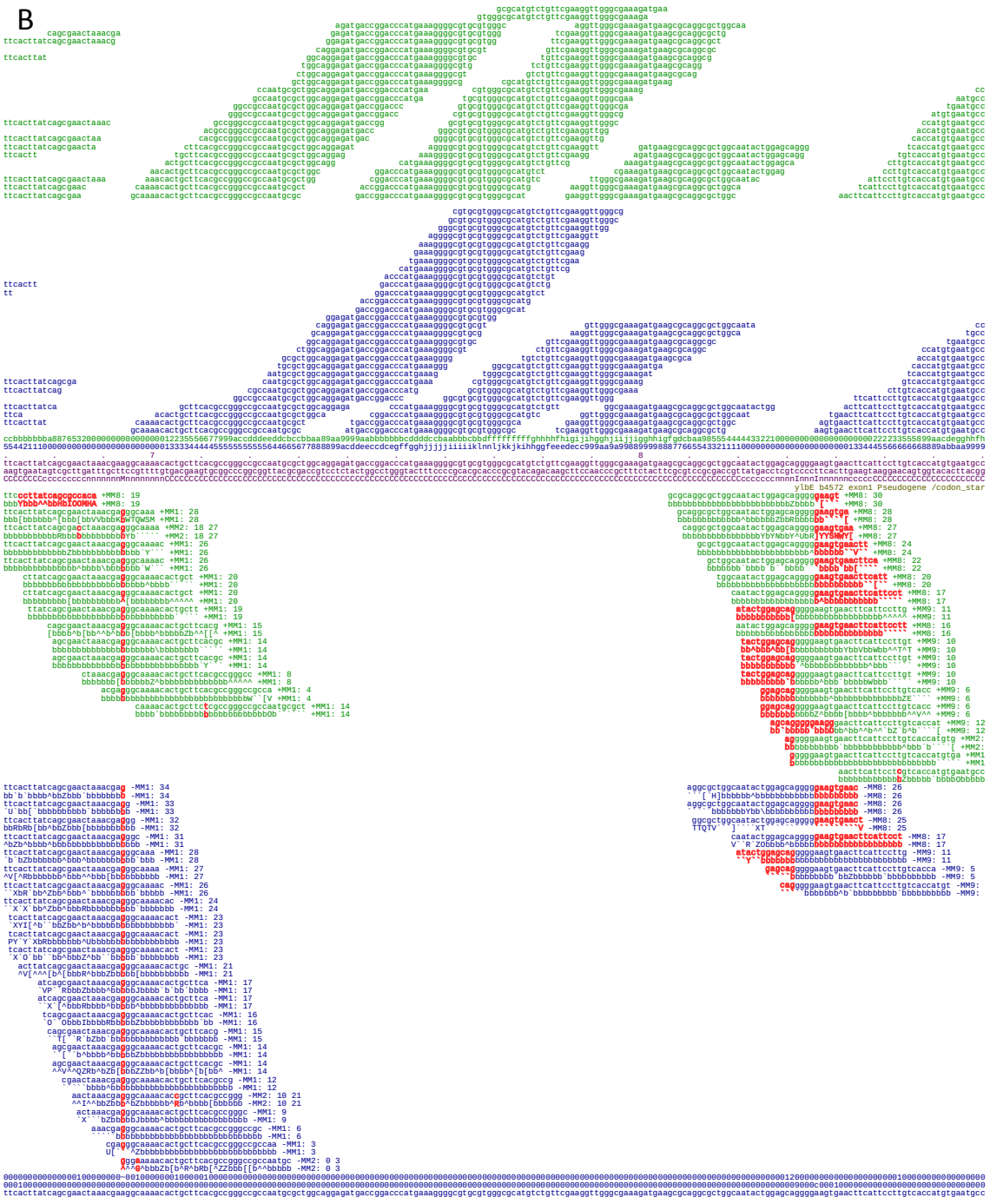


Figure 1: An example of call and presentation of a mutation and a single-nucleotide insertion. A: an excerpt of the html formatted list of “interesting positions to inspect”. In panel 547 there is a clustering of mutation reads at position 694 and clustering’s of indel reads at positions 832 and 836. No matching reads spans these positions. Clicking either link in the html file opens panel 547 in a separate window. B: a section of panel 547 (from base 670 to 870) (use zoom in to view details). From top to bottom: matching forward reads (green), matching reverse reads (blue), statistics and reference sequence, forward mutation & indel reads (green with un-

matched nucleotides in red), reverse mutation & indel reads (blue with unmatched nucleotides in red, statistics and reference sequence. The quality of each nucleotide is shown for mutation and indel reads as a separate line underneath. “b” indicates highest quality, other signs and letters indicate a lower quality. At pos 694 it is observed that i) no matching reads spans that position, ii) many mutation reads and in both directions suggest that an “a” is changed to a “g” and iii) these reads are staggered, i.e. they start at different nucleotides and are therefore not clonal reads. At pos 832 and 836 it is observed that i) no matching reads spans that position, ii) many indel reads in both directions marks either side of the gggg sequence and iii) these reads are staggered. Inspection of these reads reveals that most reads will match if an extra g is inserted in the gggg sequence. Scattered around are a few mutation and indel reads, caused by sequencing errors. These positions lack some of the observations i) to iii) and the conflicting nucleotides are occasionally also of low quality.

The R2R strategy

Most programs use memory demanding indexing strategies to find the best match between a read and the reference sequence and try to align both perfectly matching and partially matching reads at the same time and they result in so-called assembly files that require yet another program to analyze and visualize.

R2R uses a different strategy, which can shortly be described as:

- All possible n-mer sequences from the reference sequence are annotated with chromosome, position, and direction and then sorted alphabetically after sequence. n is the length of the sequence reads.
- All repeat sequence, positions where n or more bases are found elsewhere in the reference sequence one or more times, are marked
- Reads are quality-filtered and sorted alphabetically after sequence.
- Matching reads are mapped.
- The remaining pool of reads are analyzed and separated in “mutation reads” with less than n deviations from the reference sequence (n is user defined); in “indel reads” where at least m bases match the reference sequence from either the 3’ or from the 5’ end (m is user selected); or in orphan reads (no matches found). Indel reads are not used by leading programs and are the reason for the high sensitivity of this program.
- For Paired end analysis all possible mappings of a pair are evaluated.
- All mapped reads are sorted after chromosome and position in the reference sequence.
- All data are piped to call the “interesting sites” and to visualize the reads together with reference sequence, annotations, marked repeats and user defined annotations in html files readily readable with a standard internet browser.

The quality of sequence is used filter reads before and/or after mapping of matching reads and further a lower quality-limit of bases called as mutations can be set. The quality of reads can be shown for all mutation- and indel-reads.

R2R relies heavily on disk writing and reading, thus the r/w speed influence execution times. Extensive memory is only used for sorting by system commands.

What you need.

Operating System & Perl: R2R is written in simple PERL and calls only standard libraries and a few system calls. It is therefore easy to use across platforms and work easily well on Windows, any Unix/Linux, and Mac OS provided that PERL is installed.

Command line interface: see Appendix B if you are not familiar with using command line interface under Windows.

Internet browser: The resulting html files are best viewed with an internet browser; newer MS Internet Explorer's and Mozilla Firefox's works well, older versions and other browser's may or may not cause problems.

Hardware: R2R has a very low memory requirement, most scripts use less than 100 MB ram, except when they utilize system sort, then they will use available memory. R2R is greedy for disc space and a fast disc-access will improve speed.

Texteditor: You will benefit from having a nice text editor at hand for editing the .ini files. Use your favorite editor; however Windows users should get the free Metapad editor @ http://liquidninja.com/metapad/ if you are stuck with MS Notepad. Remember to save .ini files as text only should you prefer to use a high-end text program.

Data and reference sequence: Your data sequence should be the Illumina format or in standard FASTQ format. Your reference sequence should be in genbank format.

Running R2R scripts.

The R2R scripts are executed from a command prompt. This may be new land for some Mac and Windows users, but don't worry. Use the appendices and get a little help from a trained Linux user and you'll be there. For most scripts: all parameters are set in some .ini files and these settings are copied into a report file accompanying each run.

All result files are overwritten by any re-analysis of the same dataset!

Edit the .ini files with your favorite text editor and save as txt format. Make sure that your editor doesn't add ".txt" to - or delete the ".ini" part of the file name!!! For each line, the "$xxx = " part are Perl variables, do not change. The second part may have from 0 to many spaces, not tabs, before the value. ***This value may contain letters, digits, underscores, colons, spaces, dashes, slashes, backslashes and periods, nothing else.*** A semicolon or a tab should follow the value. Each line may further carry some comments. In-files can be given as filenames alone if they are placed in the working folder, else by relative or full path and filenames according to the operating system used.

For an example a full path looks like:

`$fa = D:\Solexa data\Genomes E coli\E coli S88\pECOS88;` under Windows and:
`$fa = ~/Solexa_data/Genomes_E_coli/E_coli_S88/pECOS88;` under Mac/Unix/Linux.

And a relative path looks like:

`$fa = .\E coli S88\pECOS88;` under Windows and:
`$fa = ./E_coli_S88/pECOS88;` under Mac/Unix/Linux.

if you are working in "`Genomes E coli`" or:

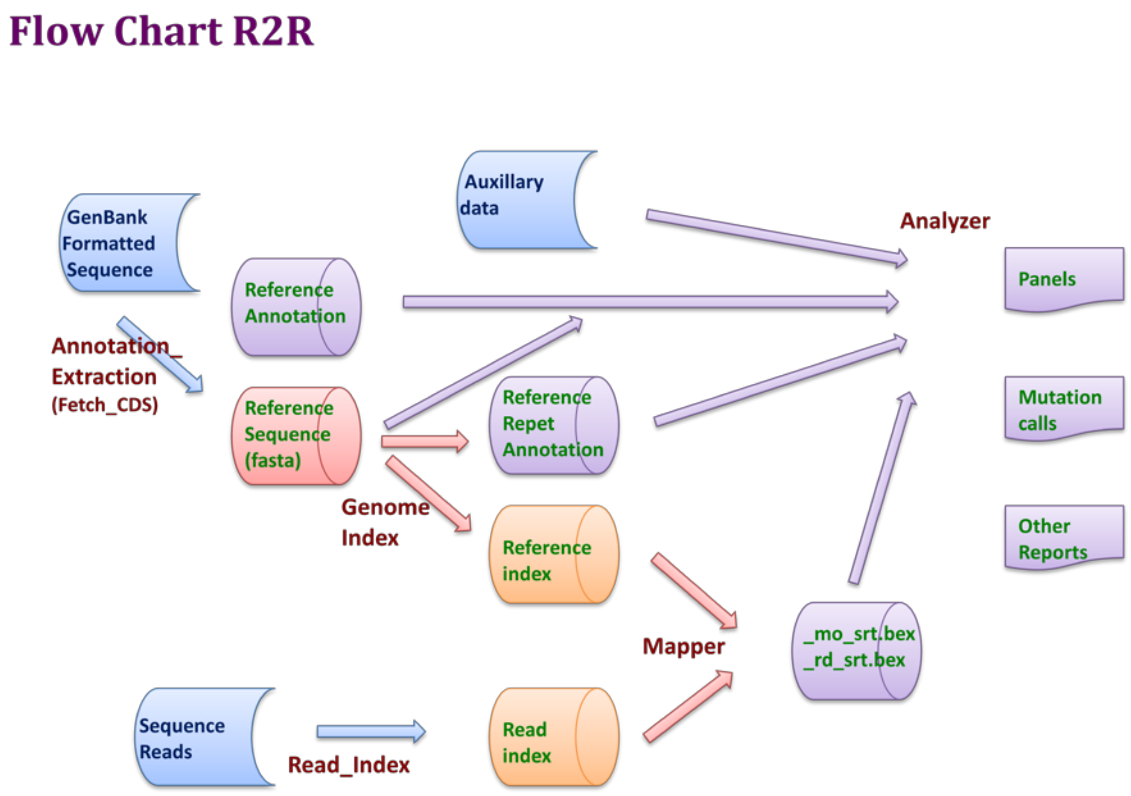
`$fa = ..\E coli S88\pECOS88;` under Windows and:
`$fa = ../E coli_S88/pECOS88;` under Mac/Unix/Linux.

if you are working in sub-folder under "`Genomes E coli`".

In my experience Mac/Unix/Linux like paths works also under Windows.

Mac/Unix/Linux systems are not fond of spaces in folder and file names; uses underscores instead or escape the spaces.

Overview – flow of data



From a GenBank file to sequence file and annotation file

If the GenBank file does not contain the translated protein sequences - R2R_Fetch_CDS.pl

Many GenBank files have omitted the translations of proteins. Look into your GenBank file after something like:

```
     CDS             join(5075..5386,1..51)
                     /locus_tag="phiX174p03"
                     /function="capsid morphogenesis"
                     /codon_start=1
                     /transl_table=11
                     /product="B"
                     /protein_id="NP_040705.1"
                     /db_xref="GI:9626375"
                     /db_xref="GeneID:2546405"
                     /translation="MEQLTKNQAVATSQEAVQNQNEPQLRDENAHNDKSVHGVLNPTY
                     QAGLRRDAVQPDIEAERKKRDEIEAGKSYCSRRFGGATCDDKSAQIYARFDKNDWRIQ
                     PAEFYRFHDAEVNTFGYF"
     gene            51..221
```

Do you find the /translation= " tag? If not try to run the script **R2R_Fetch_CDS.pl** with the GenBank file as an argument. This script identifies GI numbers within a CDS block in a standard GenBank formatted file. Then a query is send to NCBI and hopefully a protein sequence is received. This sequence is added as a /translation="xxxx section in a new file. This translation is only needed by the R2R_Annotation_Extraction.pl script to extract protein sequences for the annotations. Expected runtime depends mostly on your internet connection and the NCBI server.

Separating the content of a GenBank formatted reference sequence - R2R_Annotation_Extraction.pl

Run the script:

R2R_Annotation_Extraction.pl and give the infile, when prompted.

The infile is the full name of the Genbank formatted file. The infile may also be given as a command-line argument.

Resulting files are infile.cds and infile.fa.

The .cds files is used to pass annotation information to the R2R_Analyzer.pl.

The .fa file contains the fasta formatted sequence and is used for R2R_Genome_Index.pl; R2R_Mapper.pl; and R2R_Analyzer.pl.

Runtime: a few seconds for a standard bacterial genome.

From FASTA sequence file(s) to a genome index file - R2R_Genome_Index.pl

How to:

1. Create / copy a file Genome_Index.ini and place it in same folder as the fasta formatted DNA sequence file.

2. Edit the Genome_Index.ini file after this template:

$chr = chr1.fa	#	Name of fasta file with DNA sequence, one line for each chromosome [1)]
$start_no = 1	#	Number of first nucleotide [2)]
$chr = chr2.fa	#	Name of fasta file with DNA sequence, one line for each chromosome [1)]
$start_no = 1	#	Number of first nucleotide [2)]
$chr_end = y	#	End of chromosomes, no blank lines allowed after $chr lines
$name = E.coli_wt	#	Name of project, choose a descriptive name like strain/sample
$rd_length = N	#	Length of reads, if unknown count the number of bases of one line in the sequence data file [3)]
$index_dublicates = y	#	Create a trace for marking repeat sequence

[1)] The term "chromosome" refers actually to any division of your reference sequence into smaller contiguous fragments of your choice. If you have more than one chromosome, i.e. a bacterium with plasmid(s), more than one chromosome or several contigs or you have a eukaryote; all "chromosomes" must be indexed at once. Include also the mitochondrial sequence of eukaryotes I possible, since many reads usually are mitochondrial. The program must find one file for each chromosome with the indicated file name.

[2)] GenBank standard is "1"; UCSC genome browser use "0"; if only a section of a large sequence is given then give the coordinate of the first nucleotide to ensure match with annotation and other data. One $start_no is given for each $chr and in the same order, i.e. first $start_no applies to the first $chr.

[3)] From v. 0.950 reads of different lengths can be mapped with one index. $rd_length should then be the length of the longest reads.

3. Run the script:
   R2R_Genome_Index.pl
   with no arguments. Arguments are given by the Genome_Index.ini file.

This script generates three files, starting with $name (here E.coli_wt) and the extensions:

.srt_seqs	A large file containing the genome index

.discarded_seqs	All sub-sequences containing non-[actgACTG], for instance “n” are discarded and listed here.
.GI_report	Auxillary information and statistics

and one additional file for each chromosome with the extension:

.repeat.csv,	here chr1 .repeat.csv and chr2 .repeat.csv

with tab delimited regions for marking repeat sequences.

Runtime: expect 1-2 minutes per MB sequence.

Background:

The .srt_seqs file contains two lines for each base in the genome given to the script as one or more $chr files; one line for each strand. Each line contains in a tab separated format:

A sub-sequence with the length given by the $read_length variable

Chromosome name; here chr1.fa or chr2.fa

The coordinate of the first base of that sub-sequence

Strand (+ / -)

Repeated sequences further contain each additional occurrence in the format:

Chromosome name

Coordinate

Strand

The genome index is sorted alphabetically after sub-sequences.

The .repeat.csv file(s) simply contains one line with begin and end positions for each sequence of length $rd_length, or longer, found two or more times in the entire genomic sequence as given in the $chr file(s).

This information will eventually be shown along your sequence alignments and together with mutation calls and other statistics.

Extraction of reads from FastQ or Illumina files - R2R_Read_Index.pl

Sequence data may be obtained in several formats of which R2R reads:

- The Illumina seqpre sequence file combined with either the qcal or the qraw quality files.
- The Illumina FASTQ like sequence.txt sequence file
- Genuine FASTQ formatted sequences
- The Illumina _export.txt format is no longer supported. This and other text-based format must be converted to the FASTQ format.

See: http://en.wikipedia.org/wiki/FASTQ_format for further descriptions of FastQ.

From v. 0.950 R2R handles reads of different lengths. One or two (for paired data) of datafiles may be extracted and analyzed at a time, if you have more files you may concatenate them before analysis:

"`type file1 file2 file3 > file_all`" from a Windows Command Prompt or: "`cat file1 file2 file3 > file_all`" from any Unix/Linux/Mac prompts).

How to:

1. Create / copy a file R2R_read.ini and place it in same folder as the fasta formatted DNA sequence file.
2. Edit these variables in the R2R_read.ini file (other variables may be used by R2R_Mapper.pl):

$fid_fq	#	file with FASTQ Reads. [1)]
$fid_sp	#	file with seqpre- Reads. [1)]
$fid_ql	#	file with qual of Reads (either qraw or qcal files) [1)]
$fid_sp_1	#	If Paired End sequencing: file_1 with seqpre- Reads. [1)]
$fid_ql_1	#	If Paired End sequencing: file_1 with qual of Reads. [1)]
$fid_sp_2	#	If Paired End sequencing: file_2 with seqpre- Reads. [1)]
$fid_ql_2	#	If Paired End sequencing: file_2 with qual of Reads. [1)]
$fid_core	#	Name of experiment, sample name, strain name; used for naming out files
$qbase	#	Base number for Quality, for Illumina raw & cal use 64, for real FASTQ use 33. [2)]
$qlength	#	Length of read, that is considered for quality evalutation. [2)]
$qlimit	#	lower average quality for the first $qlength bases. [2)]

Notes:

[1)] Either ($fid_fq) OR ($fid_ql and $fid_sp) OR ($fid_ql_1 and $fid_sp_1 and $fid_ql_2 and $fid_sp_2) should be filled with filenames. Leave other variables empty.

[2] It is optional to filter for poor quality at this stage. It is recommended to filter either while indexing (this will filter matching and mutation reads) or while analyzing reads (will filter mutation reads only). Filtering can also be done in combination i.e. a course mesh here and a finer mesh later. $qlength is set to 1 and $qlimit is set to 0 for no filtering. $qlength is suggested to be set at ~80 % of readlength for filtering. The effect of filtering is evaluated by inspecting the fraction of reads removed reported by the .Read_report.txt file. With FASTQ based on ASCII value 33 most good bases will have a capital letter in the range A → I as their quality. With Illumina qualities based on ASCII value 64 most good bases will have a lower case letter a → h as their quality, deviations may occur however. An ascii table is included in the appendices.

3. Run the script:
   R2R_Read_Index.pl
   with no arguments. Arguments are given by the R2R_read.ini file

The script generates files with $fid_core as a filename and the following extensions:

.readno_srt: reads in same order as in infile, used by the R2R_mapper.pl for saving un-mapped reads.

.srt_reads: alphabetically sorted and filtered reads in a tab separated format:

- The sequence in lower case
- The quality string
- A read number assigned by the order of the original sequence file, starting with either SR (Single Read) or PE (Paired End read), [for future use].
- Read-length

.lq_reads.fq: unsorted reads filtered away in fastq format. Even in the absence of quality filtering any reads containing ambiguous sequence, i.e. non-[acgt] characters will be filtered away. This file may be used for further analysis with other programs.

.Read_report.txt: report file with general information including 1) quality settings; 2) runtime information; 3) read statistics; and 4) a copy of R2R_read.ini.

Runtime: expect 1-3 minutes per million reads.

Mapping reads to the reference sequence - R2R_Mapper.pl

R2R maps the reads to the reference sequence in two rounds. In the first round all perfectly matching reads are mapped to the position(s) where they match the reference sequence. Remaining reads are analyzed in the second round for 1) matching with less than m mismatches (mutation reads) and 2) for matching with at least n bases starting at either end (indel reads).

How to:

1. Edit these variables in the R2R_read.ini file (other variables to be used by R2R_Read_Index.pl):

$reference_in	#	The .srt_seqs file with reference sequence index generated by the genome_index0x.pl script.
$fid_core	#	Name of experiment, sample name, strain name; is now used for identifying the out files from rd_index0x.pl.
$qbase	#	Base number for Quality, for Illumina raw & calibrated use 64, for real FASTQ use 33. [1)]
$ra_qlength	#	Length of read, that is considered for quality evalutation. [1)]
$ra_qlimit	#	lower average quality for the first $qlength bases, suggested values 28 – 32. [2)]
$mut_qlimit	#	lower cut-off quality for – at least one – of the changed base(s), suggested value 35 or inspect the .Morph-report.txt to find a cut-off value.
$mut_max	#	Maximum number of mutations to look for before the read is passed on to InDel analysis; suggest 2 or 3 for shorter reads. Experiment with higher values for longer read-lengths. Reads with a more than $mut_max mis-matches will be classified as InDel reads if > $unik_treshold bases match at either 3' or 5' end. Currently 7 is the maximal value of $mut_max.
$init_treshold	#	Min no of 5' matching nt's before searching mut's, should be < ½ of $rd_length to find all single-mutation reads. $init_treshold may further have a high impact on execution time and should be in the range of 8 to 12, depending on the number of reads to analyze.
$unik_treshold	#	Min no of 5' matching nt's before searching unique InDels. $4^{\$unik\_treshold}$ should be >> twice the size of the total genome or inspect the .Morph-report.txt to find a cut-off value. For instance $4^{12} \sim 16x10^6 >> 2x4.7x10^6$ (E. coli genome) and $4^{18} \sim 64x10^9 >> 2x3x10^9$ (Human genome). I find $unik_treshold = 15 suitable for *E. coli*.
$PE_reads	#	Set to "y" for Paired End analysis; else leave empty.

Notes:

[1] It is optional to filter for poor quality at this stage. It is recommended to filter either while indexing (this will filter matching and mutation reads) or while analyzing reads (will filter mutation reads only). Filtering can also be done in combination i.e. a course mesh here and a finer mesh later. $ra_qlength is set to 1 and $ra_qlimit is set to 40 for no filtering. $ra_qlength is suggested to be set at ~80 % of readlength for filtering. The effect of filtering is evaluated by inspecting the fraction of reads removed. This is reported during execution and post-execution in the .Morph _report.txt file. With FASTQ based on ASCII value 33 [!] most good bases will have a capital letter in the range A → I as their quality. With Illumina qualities (base 64 [@]) most good bases will have a lower case letter a → h as their quality, deviations may occur however. An ascii table is included in the appendices.

2. Run the script:
   R2R_Mapper.pl
   with no arguments. Arguments are given by the R2R_read.ini file

The script generates files with $fid_core as the filename and with the following extensions:

.Morph_report.txt containing information on the analysis

.errors.txt containing error informations.

.orphans.fq containing orphan reads; i.e. reads that could not be mapped in fastq format: may be used for testing with different settings or with other programs.

.mut_discarded containing "discarded" mutation reads with the discarding reason. Experimental and FYI only. Since all reads are analyzed in both directions (complementary reads have a "c" extension to their read number) the other direction may be found as a mapped mutation/indel read.

and one folder for each chromosome containing files with these extensions:

_mo_srt.bex containing all mutation/indel reads sorted according to position.

_rd_srt.bex containing all matching reads sorted according to position.

Background:

The main steps in the mapping are:

1. Mapping all perfectly matching reads to the reference sequence. Since reads and reference sequence is indexed the algorithm is simple and this goes very fast.
2. Sorting the matching reads by the "sort" function of the operating system.
3. An optional quality filtering of all remaining reads, addition of the complement-reverse sequence of all remaining reads, and re-index.
4. Mapping remaining reads after the criteria: 1) less than $mut_max mismatch (i.e. "mutation reads") and 2) at least $unik_treshold matching bases (i.e. "indel reads"). This part is slower and a well selected quality filtering will both save computing time and help to a better sig-

nal/noise ratio, since most non-matching reads with reduced quality are caused by sequencing problems anyway.

5. Removal of duplicates. Single end reads are mapped to their first occurrence only in case they map to repeated sequence. If a pair has one read mapping to a unique sequence and the other to a repeated sequence then priority is given to map the mates in proper distance to each other.
6. Sorting the mutation/indel reads by the “sort” function of the operating system.

Runtime: highly dependent on the fraction of perfectly mapping reads and mutation/indel + orphan reads. Scales ~log-linear with an additive function of (size of reference genome and number of reads). Expect ½ to 1 hour for ~10 million reads mapped to a ~5 million bp genome using 1 cpu.

Analysis of the mapped reads - R2R_Analyzer.pl

The R2R_Analyzer.pl does a number of things:

1. Produce HTML based out-files with matching and mutation reads aligned to the reference sequence. This alignment may also include annotated features, information from paired end sequencing and (in future) other auxiliary information. The HTML files are loaded in a standard web-browser (Internet Explorer, Mozilla Firefox etc.) for visualization.
2. Produce a list of "sites to look at" with a probability of being sites deviating from the reference sequence acc. to filtering set by the user. The user can balance between "getting all simple mutations and few false positives" and "getting all identifiable mutations and many false positives". This list comes in HTML format with links to the relevant position in the alignments created.
3. Produce coverage statistics and auxiliary information.

The R2R_Mapper.pl generated one folder for each chromosome as defined in the Genome_Index.ini file. Use R2R_Analyzer.pl to analyzes each of these folders separately. R2R_Analyzer.pl is most conveniently executed from within each of these folders.

How to:

1. Edit these variables in the Eval_Reads.ini file:

$fid_core	#	Name of experiment, sample name, strain name; is now used for identifying the out files from R2R_Mapper.pl
$repeat_csv	#	Name of file tab-delimited regions of repeats in the reference sequence. Created by R2R_Genome_Index.pl
$fa	#	Name of file with reference sequence in FASTA format, this file must be identical to the relevant $chr variable of the Genome_Index.ini file
$anno	#	Name of file with annotations generated by for instance R2R_Annotation_Extraction.pl ending in .cds. Only one of $anno or $anno _bed should be given.
$snp_bed	#	A file with SNP's may be given here. No R2R scripts currently generates this. UCSC browser ??. Format in Appendix D: File-Formats.
$qbase	#	Base number for Quality, for Illumina raw & calibrated use 64, for real FASTQ use 33. See notes for the R2R_read.ini file.
$start_no	#	Number of first base in the reference sequence. See the comments in the description for R2R_Genome_Index.pl.
$print_reads	#	Do you want the matching reads printed? Set as either "y" or ";". This does not influence read statistics nor mutation / indel calls.

$print_read_density	#	Do you want the density lines of matching reads printed? Set as either “y” or “;”. If “y” one line for each direction indicating how many matching reads covers this position.
$print_mut_quality	#	Do you want the quality of mutation / indel reads printed? Set as either “y” or “;”. If “y” a string indicating the quality of each base is given in a separate line. See Fig. 1. This quality is unchanged from what was given in the in-files to R2R_Read_Index.pl.
$panel_width	#	Width of each panel in base’s. Suggested values are 100, 500 or 1000 for normal sequence, 100 or less for mitochondria and plasmids with very high coverage. More than 4000 may cause problems for web-browsers.
$seq_overlap	#	The minimal overlap between two consecutive reads to confirm overlapping sequence reads. In the example in Fig. 1 the insertion of a G happens in a GGGG sequence. A $seq_overlap value of 5 ensures that no false overlapping matching reads will be called. Suggested value 5 to 10 for short reads. $seq_overlap bases are removed from each end of the matching read in the reads statistics and in calculating whether a mutation / indel should be called with the $mut_freq_min value (see below).
$min_pr	#	Print all or some of the matching reads? If 0, all matching reads are printed, if 1 or higher, the next read must start at least $min_pr positions later than the last printed read to become printed. This does not influence read statistics nor mutation / indel calls. A value of 1 prevents piles of identical reads to be printed.
$ind_min	#	Minimum number of InDels to be reported. Suggested value at least 2. Increase for noise reduction.
$mut_min	#	Minimum number of Mutations to be reported. Suggested value at least 2. Increase for noise reduction.
$mut_freq_min	#	There must be at least ($mut_freq_min * matching reads) of mutation / indel reads to be reported. Suggested values are 1 to 2 for haploid, 0.1 - 0.3 for diploid. See also $seq_overlap above.
$ID_limit	#	The ratio of qualities of all bases left of InDel point to the right of this point must max deviate $ID_limit from 1. Suggested values: > 1 to see nearly all InDel points, 0.2 seems to separate real InDel points from noise well; this filter is still experimental.
$no_mut_in_repeats	#	Set as either “y” or “;”. If “y” (suggested value) the mutations and indels located in repeated sequence will be omitted from the mutation report.
$check_translation	#	Set as either “y” or “;”. If “y” the annotated translation will be checked and errors reported in the .Report_AUX.txt report file.

$read_dist # Minimum number of spaces between each read in the panels. Suggested value 8.

$PE_reads # Set as either “y” or “;”. “y” for analyzing Paired End reads

The parameters below are set to report unusual concentrations of PE reads, that might indicate chromosomal breaks or rearrangements. This section is ignored unless $PE_reads is set.

$PE_dist # Average distance for pairs in unbroken sequence; Look in the "Iterations ... part of the "...Morph_report.txt" report after mapping for this value: "Mean distance:" / last itaration

$PE_dist_var # Look in the "Iterations ... part of the "...Morph_report.txt" report after mapping for this values: "SD of Mean distance:" / last iteration

$PE_short_max_dist # If > $PE_short_max_dist between $PE_short_reads_in_dist, then report; indicate a **low conc** of normal PE reads.

$PE_short_reads_in_dist # See above.

$PE_long_min_dist # If < $PE_long_min_dist between $PE_long_reads_in_dist, then report; indicate a **high conc** of long PE reads.

$PE_long_reads_in_dist # See above.

$PE_unpair_min_dist # If < $PE_unpair_min_dist between $PE_unpair_reads_in_dist, then report; high conc of unpaired PE reads.

$PE_unpair_reads_in_dist # See above.

2. Run the script:
   R2R_Analyzer.pl
   with no arguments. Arguments are given by the Eval_Reads.ini file

The script generates files with $fid_core as a filename and the following extensions:

.Report_AUX.txt: Reports auxiliary information such as:

Translation errors – disagreement between annotated protein and nucleotide sequence acc. to the standard genetic code. Requires that $check_translation is set.

General and statistical information

List of unconfirmed nucleotides: Positions where no reads covers with extension of $seq_overlap bases.

Information for Gamma-statistics (number of bases with indicated number of matching reads starting at / covering the base in each direction). Not documented further.

A copy of the Eval_Reads.ini file.

.Report_Reads.csv: Number of mapped reads per panel. Tab separated: Panel number; Reads; [1]; [R]. “1” and “R”indicates that repeat sequence is present in this panel. This file may be used for read frequency analysis in the search for duplications or for marker frequency analysis. These data may be used to replace the example data in the accompanying “Read Frequency Analysis.xlsx” file. Replace the first four blue columns with data from “.Report_Reads.csv”; set chromosome data and panel_with in the yellow marked columns.

.Index.htm: html formatted file with links to each panel, based on coordinates or name of genes.

.Report_Mutations.htm: html formatted with links to panels that may contain mutations or breaks. This is most likely your work horse for chasing mutations and break points. See more in the chapter: Finding mutations.

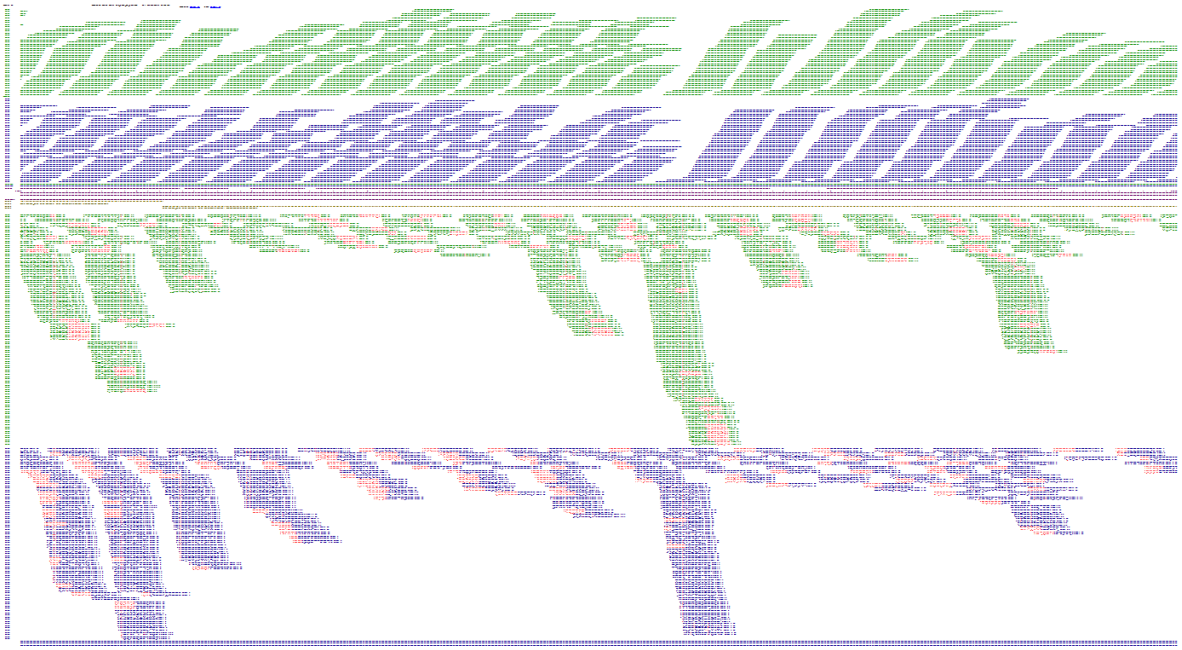
.Report_Unconfirmed_bases.htm: html formatted with links to panels that contains bases not confirmed by an overlapping matching read, the sites may or may not be covered by mutation or indel reads. Indicates deletions and other deviations.

and a sub folder named “html” and containing one .htm file for each panel. These file contains alignment of the mapped reads with the reference sequence and auxiliary information as selected. The files may be opened directly or using links in any of the above .htm files for inspection of interesting regions of the sequence.

Finding mutations

Layout of Panels

The following refers to the html formatted panels. The overall layout of each panel is as shown in this example of 1 kb wide panel:



From top to bottom the standard panel contains:

- links to previous & next panel, a stamp
- forward perfectly matching reads in green, reverse perfectly matching reads in blue
- a section with sum of matching reads, repeat marks, a ruler, reference sequence, a combined coverage/repeat/mutation call/InDel call/ line, annotations and other information.
- forward Mut/InDel reads in green, reverse Mut/InDel reads in blue, deviations from reference sequence marked in red. Each read contains by default two lines: the sequence and the quality.
- summary of Mut/InDel reads and the reference sequence.
- If Paired End data are analyzed there will further be a section with these data below the Mut/Indel section.

Many of these elements are optional, depending on the selections in the Eval_Reads.ini file. The summary lines shows the numbers of matching reads / Mut reads / InDel reads using this notation:

0 → 9 indicate 0 → 9; a → z indicate 10 →35 and ~ indicates > 35.

A few hints:

1. Panels with a high coverage may create difficulties in some browsers. In that case select shorter panels, i.e. 100 or 200 bp wide instead of 1000 bp. Most browsers will have problems showing more than 4000-some characters wide panels.
2. The zoom function in some browsers may be helpful to get an overview of the panel.
3. Excerpt figures – like those used for illustrations in this manual – can be created this way: Copy and paste the entire panel from browser to an empty Word doc. Select all, set font size to 1. Set page orientation to landscape and adjust margins if needed. Use the "Extend a selection or block" block copy function in Word. This function is not well described, however place the cursor in one corner of the desired area; then press Ctrl_Shift+F8; then - while holding the Shift down - move the cursor to the opposite corner with arrow keys or a mouse click. This marks an rectangle, when satisfied copy this and paste is elsewhere. The paste will be a block – ensure that enough blank lines are available or paste at the end of a text. Increase font size again until satisfied.

"Reading" the .Report_Mutations.htm file

R2R analyzes each individual read for match / mismatches, align the reads with reference sequence and other information. To decide whether a position is actually changed or whether a number of sequence errors have piled up at a particular place needs so far a manual inspection. The .Report_Mutations.htm file includes all positions called to be "possible positions with deviations in sequence". Both noise and signal in this file is highly dependent on the settings provided and the coverage and quality of the sequence reads.

Some experience is required to separate mutations from sequence artifacts, this is just to help get started. The examples below shows some lines from the .Report_Mutations.htm file along with a discussion and in some cases a simplified excerpt of the corresponding panel. All examples are derived from the accompanying Medium dataset (except when noted); however panels are simplified to show the point.

The headings of the .Report_Mutations.htm file and the examples reads (with reference to example 1 below): **Panel** and **Pos** are panel number and position in this panel, since panels were selected to be 1000 bp wide this translates into position 299.443 in the chromosome sequence. **Type** is either "Mutation" or "InDel". **#Mut, #Mut+, #Mut-,** are the total, on plus strand, on minus strand number of reads supporting the mutation/InDel at this position. **#Match** is the number of matching reads at this position. **ID_qual** (InDel calls only) is the fraction of all bases left of pos to all bases right of pos, i.e. a measure of quality of that InDel call. Values near 1.00 indicates are more likely to be true than values significantly different from 1 (in practice 0.9 and 1.1 so far appears to be good cut-off values). **Annotation** is the annotated information to this position.

Example 1 (a true mutation):

Panel_No	P_Pos	Mut/InDel	#Mut	#Mut+	#Mut-	#Match	ID_qual	Repeat	SNP?	Suggested mutation(s)	Annotation(s)
299	443	Mutation:	27	14	13	0				c>t (1.00)	G>D; yagR (predict

Many Mut reads, Mut reads are found on both strands and no Match reads are found, no Repeat sequence. This is most likely a true mutation.

Example 2 (simple InDel in a poly-nucleotide run):

```
Panel_No  P_Pos   Mut/InDel     #Mut  #Mut+  #Mut- #Match ID_qual Repeat  SNP?  Suggested mutation(s)      Annotation(s)

    547    832   InDel:          24     15      9      0    0.97                                          ylbE (Pseudogene
    547    836   InDel:          18     13      5      0    1.02                                          ylbE (Pseudogene
```

(Part of Fig. 1) This is a typical example of detecting indel bases in a poly-nucleotide run. In this case 4 g's turn into 5 g's. The ID_qual is very near 1.0 for both calls. Notice that the two InDel calls are shiftet with 4 bases. Notice also how the setting of $seq_overlap = 7 prevents the intruding reads from marking the bases as confirmed in the panel excerpt.

Example 3 (longer InDel):

Panel_No	P_Pos	Mut/InDel	#Mut	#Mut+	#Mut-	#Match	ID_qual	Repeat	SNP?	Suggested mutation(s)	Annotation(s)
2616	100	Mutation:	3	1	2	0				c>g (1.00) L>F;	hda
2616	100	InDel:	7	5	2	0	1.10				hda
2616	849	Mutation:	4	1	3	0				c>t (1.00)	
2616	850	Mutation:	5	2	3	0				c>a (1.00)	
2616	851	InDel:	8	5	3	0	0.99				

```
Panel: 2616    Matching reads file: hsm-3_rd_srt.bex    on: 22-08-2011: 15:02:22
                                Previous Panel: 2615    Next Panel: 2617
  +Rds
  +Rds
  +Rds
  +Rds
  +Rds
  +Rds        cgtagttcggataaggcgttcg
  +Rds        cgtagttcggataaggcgtt
  +Rds        cgtagttcggataaggc
  +Rds                                                                                                                                     ttatttgtc
  +Rds                                                                                                                               ggtacattatttgtc
  +Rds                                                                                                                              cggtacattatttgtc
  +Rds                ggataaggcgttcgcgccgcatccgacaataaaca                                                                          tcggtacattatttgtc
  +Rds            gttcggataaggcgttcgcgccgcatccgacaata                                                                      ctgcccggtcggtacattatttgtc
  +Rds        cgtagttcggataaggcgttcgcgccgcatccga                                                                        acactgcccggtcggtacattatttgtc
  +Rds        cgtagttcggataagg                                                                                     cagcaacactgcccggtcggtacattatttgtc
  +Rds        cgtagttcggataa                                                                                      gcagcaacactgcccggtcggtacattatttgtc
  +Rds        cgtagttcgg                                                                                   aaccgggcagcaacactgcccggtcggtacattat
  +Rds        cgtagttcggataaggcgttcgcgccgca                                                              ttgaaccgggcagcaacactgcccggtcggtacat
  +Rds        cgtag          gcgttcgcgccgcatccgacaataaacaccttatc                                          tttgaaccgggcagcaacactgcccggtcggtaca
  +Rds        cgt           ggcgttcgcgccgcatccgacaataaacaccttat                                       gatgtttgaaccgggcagcaacactgcccggtcgg          tc
  +Rds        cgtagttcggataaggcgttcgcgccgcatccg                                                     catgatgtttgaaccgggcagcaacactgcccggt          gtc
Anchor
  -Rds
  -Rds                                                                                                                                       atttgtc
  -Rds                                                                                                                                     ttatttgtc
  -Rds                         gttcgcgccgcatccgacaataaacaccttatcta                                                                       cattatttgtc
  -Rds                     aggcgttcgcgccgcatccgacaataaacacctta                                                                       ggtacattatttgtc
  -Rds                    aaggcgttcgcgccgcatccgacaataaacacctt                                                                       cggtacattatttgtc
  -Rds          tagttcggataaggcgttcgcgccgcatccgacaa                                                                          gcccggtcggtacattatttgtc
  -Rds        cgtagttcggataaggcgttcgcgccgcatccga                                                                      cactgcccggtcggtacattatttgtc
  -Rds        cgtagttcggata                                                                                         aacactgcccggtcggtacattatttgtc
  -Rds         gtagttcggataaggcgttcgcgccgcatccgaca                                                                 caacactgcccggtcggtacattatttgtc
  -Rds        cgtagttcggataaggcgttcgcgccgcat                                                                    agcaacactgcccggtcggtacattatttgtc
  -Rds        cgtagttcggataaggcgttcgcgc                                                                     gcagcaacactgcccggtcggtacattatttgtc
  -Rds        cgtagttcggataaggcgttcgc                                                                      ggcagcaacactgcccggtcggtacattatttgtc
  -Rds        cgtagttcgga                                                                                cgggcagcaacactgcccggtcggtacattatttg
  -Rds                 gataaggcgttcgcgccgcatccgacaataaacac                                                  ccgggcagcaacactgcccggtcggtacattattt
  -Rds        cgtagttcggataaggcgttcgcgccgcatccg                                                     aaccgggcagcaacactgcccggtcggtacattat
  -Rds        cgtagttcggataaggcgttcgcgccgcatcc                                                     gaaccgggcagcaacactgcccggtcggtacatta
  -Rds        cgtagttcggataaggcgttcgcgccgc                                                    tgtttgaaccgggcagcaacactgcccggtcggta
  -Rds        cgtagttcggataaggcgttc                                                          atgtttgaaccgggcagcaacactgcccggtcggt          gtc
  SUM+Rds     6776678776577666666667888876666644443333332000000000000000000000000000     00000000000000000000111222234446666667888778778876664457887666677
  SUM-Rds     8899abccdeeeeebbccbdeddcdba8876666666644421111000000000000000000000000     000000000000000000000001222223445667999aacddedccdddcbbaaba889987
  Repeat
      26160  .         .         .         .         .         1         .          .         .         .         .         .         .        9
              cgtagttcggataaggcgttcgcgccgcatccgacaataaacaccttatctacaacttcagaatttct     accacagaatcccatgatgtttgaaccgggcagcaacactgcccggtcggtacattatttgtc
              gcatcaagcctattccgcaagcgcggcgtaggctgttatttgtggaatagatgttgaagtcttaaaga     tggtgtcttagggtactacaaacttggcccgtcgttgtgacgggccagccatgtaataaacag

  Coverage    CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCccnnnnnnnInnnnnnnnnnnnnnn     nnnnnnnnnnMMInnnnnnccccCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
  Annot                                                      ***.L,.K,.L,.I,.E,.                                                              ***.K,.D,
  Annot                                                      hda b2496 exon1 ATP                                                              uraA b249

  +MID        cgcagttc +MM8: 17           atccgacaataaacaccttatctagcaccaggcgt +MM8     gctgccaacttacatgatgtttgaaccg +MM9: 18
  +MID        CIII=III +MM8: 17           IIIIIIIIIIIIIIIIIIIIIIII2IIIIIIIIII +MM8     IIIIIIIIIIIIIIIIIIIIIIIIIIII1 +MM9: 18
  +MID        tgtaggccgcataaggcgttcgcgcc +MM9: 18                                      gctgccaacttacatgatgtttgaaccgggca +MM9: 14
  +MID        IIIB*:I,I';I-9=43@D!6?+:;" +MM9: 18                                      IIIIIIIIIIIIIIIIIIIIIIIIIII6IIII +MM9: 14
  +MID                ggataaggcgttcgcgccgcatccgccactacaca +MM8: 25                     gctgccaacttacatgatgtttgaaccgggcag +MM9: 13
  +MID                IIIIIIIIIIIIIIIIIIIIIIII6#I;(H1"@I+ +MM8: 25                     IIIIIIIIIIIIIIIIIIIIIIIIIIIIICICI +MM9: 13
  +MID        15 25           cgttcgcgccgcatccgacatccgcagtcccccgc +MM8: 20                    acttacatgatgtttgaaccgggcagcaacactgc +MM9: 4
  +MID        15 25           ICIIIIIIIIIIIIIIIIIIIIII*I5I9II2'II +MM8: 20                    IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII +MM9: 4
  +MID        : 25              ttcgcgccgcatccgacaataaacaccttatctag +MM1: 34                   ggttcatgttgtttgaaccgggcagcaacactgcc +MM9: 8
  +MID        : 25              IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIA +MM1: 34                   "II1I(IIIIIIIIIII*III3*A$5I:II2$FI0 +MM9: 8
  +MID        M8: 18                cgccgcatccgacaataaacaccttatctagcacc +MM8: 30               cttacatgatgtttgaaccgggcagcaacactgcc +MM9: 3
  +MID        M8: 18                IIIIIIIIIII:I7IIF=II=IHFF6,:9,--,1- +MM8: 30               IIIIIIIIIIIIIIIIIIIIIIIIII=IIIIIIII +MM9: 3
  +MID                               gccgcatccgacaataaacaccttatctagcacca +MM8: 29                tacatgatgtttgaaccgggcagcaacactgcccg +MM2: 0 1
  +MID                               IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII +MM8: 29                IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII +MM2: 0 1
  +MID                                    atccgacaataaacaccttatctagcaccaggcgt +MM8               acatgatgtttgaaccgggcagcaacactgcccgg +MM1: 0
  +MID                                    IIIIIIIIIIIIIIIIIIIIIIII(I9IIDIIII( +MM8               IIIIIIIIIIIIIII=IIIIIIIII=IIGBIII= +MM1: 0
  +MID                                         acaataaacaccttatctagcaccaggcgtttaag                   gatgtttgaaccgggcagcaacactgccccggcgg +MM2: 29 31
  +MID                                         IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII=I                   IIIIIIIIIIIIIIIIIIIIIIIII&III;+:@III +MM2: 29 31
  +MID                                         acaataaacaccttatctcgcaccagtctttcacg                                        gatttggattctgacattgatttgtc
  +MID                                         IIIIIIIII%IIIIIIII"IIIIIC=$@+:I!34F                                        IIIIIIIIIIIIIIIIIIIIIIIIII

  -MID        cgtagttctgataag -MM1: 28                                                 gctgccaacttacatgatgtttgaaccgggc -MM9: 15
  -MID        IIIIIIIIIIIIIII -MM1: 28                                                 (II4IIIIIIIIIIIIIIIIIIII;IIIIII -MM9: 15
  -MID        tgtagggcggataaggcgttcgc -MM9: 18                                            gccaacttacatgatgtttgaaccgggcagcaaca -MM9: 8
  -MID        IIIIII(IIIIIIIIIIIIIIII -MM9: 18                                            IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII -MM9: 8
  -MID        cgtagttcggataaggcgtacgcgccgcatccta -MM2: 20 33                                 aacttacatgatgtttgaaccgggcagcaacactg -MM9: 5
  -MID        III;GI%IIIIIIIII5I1IIIIIIIIIIIIIII -MM2: 20 33                                 III6IIIIIIIIIIIIIIIIIIIIIIIIIIIIIII -MM9: 5
  -MID         ttagttcggataaggcgttcacgccgcatccgaca -MM2: 0 20                                   ttacatgatgtttgaaccgggcagcaacactgccc -MM2: 1 2
  -MID         $.II=IIIIIIIIIII=IIIIIIIIIIIIIIIIII -MM2: 0 20                                   IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII -MM2: 1 2
  -MID        cgtaggccggataaggcgttcgcg -MM9: 17                                                 ttacatgatgtttgaaccgggcagcaacactgccc -MM2: 1 2
  -MID        IBIIIIIIIIIIIIIIIIIIIIII -MM9: 17                                                 ?IIIIIIIICIIIIIIIIIIIIIIIIIIIIIIIII -MM2: 1 2
  -MID        atttccctggataaggcgttcgcgc -MM9: 17                                                 tacatgatttttgaaccgggcagcaacactgcccg -MM9: 8
  -MID        IH=*IIIIIIIIIIIIIIIIIIIII -MM9: 17                                                 IIIIIIIIIIIIIIIIIIIIII)IIIIIIIIIIII -MM9: 8
  -MID        agtgccctggataaggcgttcgcgccgcatcc -MM9: 10                                          tacatgatgtttgaaccgggcagcaacactgcccg -MM2: 0 1
  -MID        I!@IIIIIIIIIIIIIIIIIIIIIIIIIIIII -MM9: 10                                          =IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII -MM2: 0 1
  -MID                     aggcgttcgcgccgcatccgacatccggagtgcca -MM8: 23                             atgatgtttgaaccgtgcagcaacactgcccggtc -MM1: 15
  -MID                     III%IIIIIIIIIIIIIIIIIIIIIIIIII'III -MM8: 23                              H':II5II0GIIII0IIII3IIII)IIIIIII2II -MM1: 15
  -MID                           tcgcgccgcatccgacaataaacaccttatctagc -MM2: 33 34                         gtttgaaccgtgcagcaccactgcccggtcggtac -MM2: 10
  -MID                           IAIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII -MM2: 33 34                         #I=III55IIFIII%II2IIIIII$I7IIIIIIII -MM2: 10
  -MID                             gcgccgcatccgacaataaacaccttatctagcac -MM2: 31 32
  -MID                             7IIII*IIIIIIIIIIIIIIIIIIIIIIIIIIIII -MM2: 31 32
  -MID                              cgccgcatccgacaataaacaccttatctagcacc -MM8: 30
  -MID                              I'I69$II-IIIIIIIIIIIIIIIIIIIIIIIIII -MM8: 30
  -MID                                   catccgacaataaacaccttatctagcacccggcg -MM8:
  -MID                                   9I6=IIIIIIIIII%IIIIIIIIIIIIIIIIIIII -MM8:
              0100000010000000000101000000000010000000000000000000032000000000000000     0000000000450000000000000000002000000100000000101000000000000000000
              0000000220100000000000000000000000100200000000000000017000000000000000     0000000000008000010100000000000000000000000000000000000001000000
              cgtagttcggataaggcgttcgcgccgcatccgacaataaacaccttatctacaacttcagaatttct     accacagaatcccatgatgtttgaaccgggcagcaacactgcccggtcggtacattatttgtc
```

The report says that two positions at 100 and 851 have clusters of InDel reads, inspection of Panel 2616 shows that no reads match in the region in between. The two "ends" – in red – does not splice to anything meaningful. This marks the intended *hda::cat* replacement in this strain. Notice that InDel reads are on both strands and that ID_qual is very near to 1. Such positions are often associated with mutation calls as well.

Example 4 (sequence problems, old example):

```
Panel  Pos  Type   #Mut   #Mut+   #Mut-  #Match  ID_qual Rep.  Annotation
   88  149  InDel:   12      12       0       7     1.55       fruR (DNA-

gcgaagcaataccgtgtgagcgacaaaaccg                                  ttaccacccgaac
gcgaagcaataccgtgtgagcgacaaaacc                        ggtgcgtgagcacaattaccacccgaac
            cgtgtgagcgacaaaaccgttgaaaaagtcatggc                  aattaccacccgaac
gcgaagcaataccgtgtgagcgacaaa         aaagtcatggctgtggtgcgtgagcacaattacca
gcgaagcaa           gcgacaaaaccgttgaaaaagtcatggctgtggtg        caattaccacccgaac
gcgaagca          agcgacaaaaccgttgaaaaagtcatggctgtggt        acaattaccacccgaac
         taccgtgtgagcgacaaaaccgttgaaaaagtcat           gtgagcacaattaccacccgaac
    agcaataccgtgtgagcgacaaaaccgttgaaaaa         gtggtgcgtgagcacaattaccacccgaac
gcgaagcaataccgtgtgagcgacaaaac               ggctgtggtgcgtgagcacaattaccacccgaac
gcgaagcaataccgtgtgag             aaaaagtcatggctgtggtgcgtgagcacaattac        c
cdeddcbbccbbaa99a98aba9888877666666665431000000000000022333334688889999999 98a9
777888899a9aa888aaab9987678988888bbbb999899aaa98777776344555556655578999999aa
1         .         .         .         .         .         .         .
gcgaagcaataccgtgtgagcgacaaaaccgttgaaaaagtcatggctgtggtgcgtgagcacaattaccacccgaac
cgcttcgttatggcacactcgctgttttggcaacttttcagtaccgacaccacgcactcgtgttaatggtgggcttg
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCIccccCCCCCCCCCCCCCCCCCCCC
,A.,K.,Q.,Y.,R.,V.,S.,D.,K.,T.,V.,E.,K.,V.,M.,A.,V.,V.,R.,E.,H.,N.,Y.,H.,P.,N.

                      caaaaccgttgaaaaagtcatggctgggggggggg +MM8: 26
                      bbbbbbbbbbbbbbbbbbbbbbbbbZWbbTSZGG +MM8: 26
                      caaaaccgttgaaaaagtcatggctgggggtggtt +MM8: 26
                      bbbbbbbbbbbbbbbb^bbbbbZbbbHZIHQUAGA +MM8: 26
                       aaaaccgttgaaaaagtcatggctgggggggggtgg +MM8: 25
                       bbbbbb[bb^bbbbbbbbbb^bbbbZ[bIRVTGGM +MM8: 25
                       aaaaccgttgaaaaagtcatggctgggggggggtgg +MM8: 25
                       bbbbbb`bbbb```^`bbbbb`bb`ZR`ZJ`OAVG +MM8: 25
                        aaaccgttgaaaaagtcatggctgggggggggggg +MM8: 24
                        bbbbbbbbbbbbbb^bbbbbb^bbbbbbbbH`YTY[ +MM8: 24
                        aaaccgttgaaaaagtcatggctgggggggggggg +MM8: 24
                        bbbbbbbbbbbbbbbbbbbbbbbbZbRZ^IVUISG +MM8: 24
                         aaccgttgaaaaagtcatggctgggggggggggg +MM8: 23
                         bbbbbbbbbbb`bb`bbbbbbb`^^^IbQ`OXG`V +MM8: 23
                            cgttgaaaaagtcatggctgggggggggggggg +MM8: 20
                            bbbbb`b`bbbbb``bbbbb^bbZ^Z^Z^`[OREC +MM8: 20
                             gttgaaaaagtcatggctgggggggtgggggaa +MM8: 19
                             bbbbbbbbbbbbbbbbbbbJb^RZ^ZIXXXVMA[U +MM8: 19

00000000000000000000010000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000c000000000000000000000000000
gcgaagcaataccgtgtgagcgacaaaaccgttgaaaaagtcatggctgtggtgcgtgagcacaattaccacccgaac
```

There will be numerous cases where such an InDel points to sequence problems, in this example non-matching sequences are mostly G's (would be C's in reverse direction). Further i) the quality drops significantly at the break point from mostly lowercase b's to uppercase letters, reflected in the ID_qual being high: 1.55 and ii) all InDel reads are on the same strand. Harsher filtering on the quality of reads can somehow reduce the number of false InDel's.

Example 5 (expansion between short duplications):

```
Panel  Pos  Type   #Mut   #Mut+   #Mut-  #Match  ID_qual Rep.  Annotation
 1397  493  InDel:   14       4      10      11      0.98       fnr (DNA-binding
```

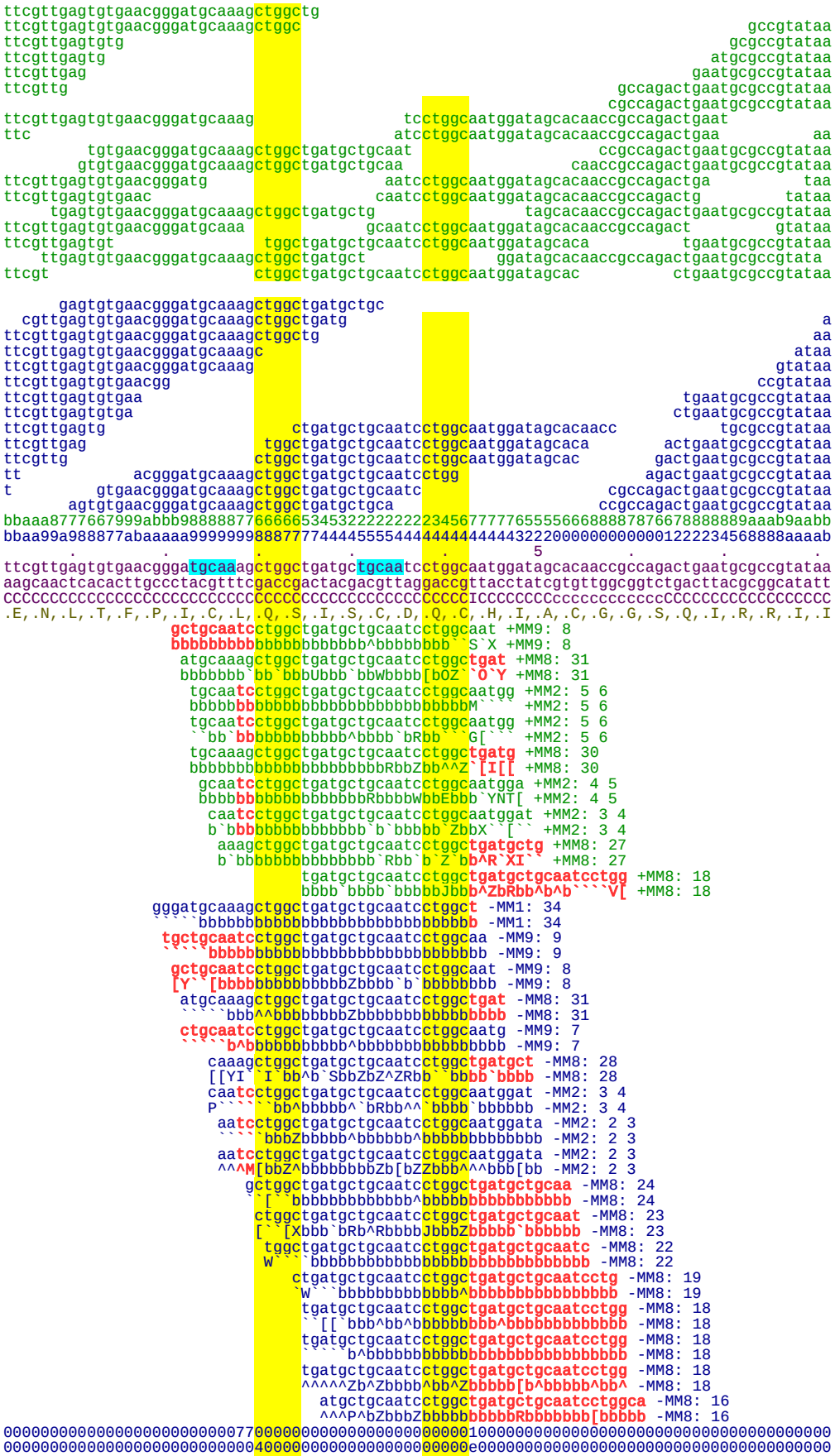
This example is more complicated – see the panel on next page. At a first glance numerous matching reads spans region called as a possible InDel. The InDel reads however are: 1) independent and some on the reverse strand, though this seems to be dominated by mutation reads and 2) generally of good quality, ID_qual is 0.98. Marking (yellow) the suggested InDel positions on the reference sequence reveals two 5 bp direct repeats:

agtgtgaacgggatgcaaagctggctgatgctgcaatcctggcaatggatagcacaaccgccagactgaatgcg

this suggests that a sequential duplication may have occurred. Changing the sequence above into:

agtgtgaacgggatgcaaagctggctgatgctgcaatcctggctgatgctgcaatcctggcaatggatagcacaaccgccagactgaatgcg

in the genomic reference sequence and repeating the entire analysis confirmed that this is the optimal solution for these data (not shown). Notice that in this case only 4 InDel reads and 4 Mutation reads marks the left side compared to 6 + 8 = 14 matching reads and thus that position was not called with the used settings. A part of the reason is the tgcaa repeats marked in blue. More sensitive settings (higher $seq_overlap diminish the matching reads and a lower $mut_freq_min will call more sites) will call such positions, but also generate more noise.



Appendix A: Installation of Perl and the R2R scripts.

Install Perl.

Most Mac, Unix and Linux users will find that Perl is already installed on their systems.

Windows users may have to install Perl themselves.

For all operating systems www.perl.com is the first place to visit and find help and instructions. Go to downloads. I have used ActivePerl, v.5.10.0 for this development, find the recent version at: www.activestate.com/ActivePerl/. Strawberry Perl http://strawberryperl.com/ is an alternative.

Install R2R.

Mac / Unix / Linux users either place the Perl scripts in their /usr/bin/ folder or in any other folder. In the latter case you will wish to add that folder to your path, see Appendix C.

These scripts have been tested less thoroughly on a Mac OS and thus some bug-fixing might be expected.

Windows users should identify the folder assigned by Perl for your Perl scripts and place the Perl scripts there. On my PC this is C:\Perl\site\bin. Alternatively Windows users can place their scripts in any folder created by them such as C:\MyPerls. Then that folder must then be added to the Windows Path variable. See "Appendix C:".

Run-times

To give an idea about expected run-times the dataset "hsm2" was analyzed in parallel on a lap-top and a medium-sized Linux-box with same settings. Analysis was done on one processor only on both systems. The hsm2 data set includes ~7 million reads analyzed against the 4.6 million bp E. coli MG1655 genome. Run-times will depend on the cache content and on other processes on the computer since a lot of data is read/written to the disc.

Table 1: Example of run-times

Script/Process	Lap top	Linux-box
R2R_Annotation_Extraction.pl (sec)	8	6
R2R_Genome_Index.pl (sec)	360	66
R2R_Read_Index.pl (sec)	424	256
R2R_Mapper.pl [1)] (sec)	570	267
R2R_Analyzer.pl (sec)	596	274
Total for 1'sample (minutes)	32.6	14.5
Total for subsequent samples (minutes)	26.5	13.3

[1)] Mapping is especially sensitive to the fraction of perfectly matching reads to remaining reads, which again is a function of the quality of reads, the experimental setup, and quality filtering settings. The performance with this set is fast compared to other similar datasets.

Appendix B: Command Prompt under Windows.

I enjoy executing the Perl Script from the Command Prompt.

I actually place a copy of a short cut to the command prompt in each working folder.

This makes it easier to follow executions, get files placed correctly, and to read error messages.

To find your first shortcut to command prompt open the "Start" menu, in accessories find the icon for "Command Prompt" and copy this to your first folder, either by "drag and drop" while Ctrl is pressed, (http://en.wikipedia.org/wiki/Drag-and-drop) or by right-click, select "copy" and so on.

Double-click to open the "Command Prompt".

You may wish to change the lay-out by right-clicking the command bar and select "Properties". Select appealing colors and increase Screen buffer size to > 1000 in the Height are my favorites. Select "Modify shortcut that started this window" on exit. Copy this shortcut further to any folder where you need it.

1'st check: to see if Perl is properly installed on your computer type "perl –v" and enter at the command prompt. If OK some 10 lines starting with "This is Perl …" appears, otherwise your find some 2 lines starting with "Perl is not recognized …"

Appendix C: Setting the path under Windows and MacOS

Windows:

This instruction is for Windows XP, may deviate for other systems. If this instruction is not helpful you need assistance from your system administrator.

Find and open “System“ in your “Control Panel”.

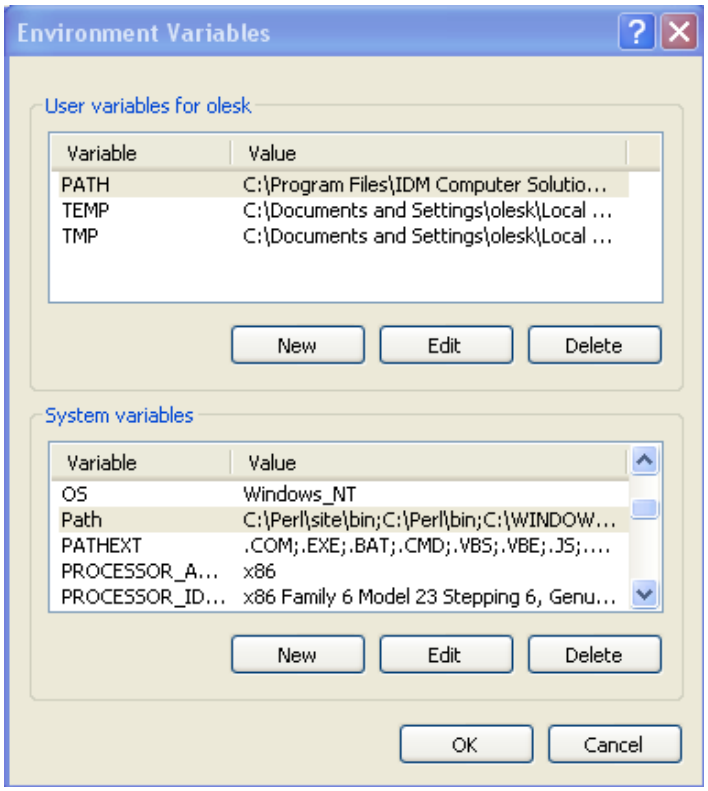
Under the “Advanced” tab open “Environment Variables”.

In either part “User variables for (user)” or “System variables” identify the path variable.

Add the path to the folder containing your Perl scripts after a semicolon. (Suggestion: copy/paste the content to a text editor like Notepad; edit; copy/paste back again).

The new path is effective after a restart of the computer.

To avoid restart you can issue the command in a Command Prompt: PATH = %PATH%;C:\MyPerls



This will add the dir C:\MyPerls to your previous path for the current session only.

Check your path by issuing the command: PATH

Windows 7: Find the Env. Variables in System -> Advanced System Settings -> Environment variables.

MacOS:

The following command may be useful to Mac users for setting the path:

From your terminal move to the folder containing your scripts, then issue the command:

```
echo "export PATH=\$PATH:$(pwd)" >> ~/.bashrc
```

```
source ~/.bashrc
```

This will add your script folder to your path. To check the effect issue:

```
echo $PATH
```

Appendix D: File formats

SNP files:

An example with selected lines from a SNP file ($snp_bed) downloaded from the UCSC browser:

#chrom	**chromStart**	**chromEnd**	name	strand	refUCSC	**observed**	class	func
chr15	**37588032**	**37588033**	rs16969162	+	T	**C/T**	single	unknown
chr15	**37588033**	**37588033**	rs5812123	+	-	**-/C**	insertion	unknown
chr15	**37589187**	**37589188**	rs8025243	+	A	**A/G**	single	unknown
chr15	**37603625**	**37603625**	rs11450909	+	-	**-/A**	insertion	unknown
chr15	**37604536**	**37604537**	rs11070216	+	T	**C/T**	single	unknown
chr15	**37605777**	**37605778**	rs7183531	+	A	**A/G**	single	unknown
chr15	**37605885**	**37605886**	rs5812124	+	T	**-/TTGT**	deletion	unknown
chr15	**37605916**	**37605916**	rs3076975	+	-	**-/GTTT**	insertion	unknown

only the information marked with **bold red** is actually used

Appendix E: A quick start tutorial

Some just like to get started and then investigate functions and settings later. If that's you try this:

Get started:

Hint: commands are found in "R2R_commands.txt" and can be issued as multiple commands or as a batch.

1. Download and un-compress R2R scripts with test-data.
   All program files are plain Perl scripts, no libraries are needed.
   http://milne.ruc.dk/R2R
   Some assistance is found in the previous appendixes: "Installation of Perl and the R2R scripts" and for Windows users also: "Prompt under MS ®Windows".

2. Locate the folder "Release_0.950" – or any update of this.

3. Start a terminal:
   Windows: copy a Command Prompt to the folder "Release_0.950", see Appendix B, and double-click this.
   MacOS: find the Applications/Utility/Terminal and start this.
   Unix/Linux: you know how to do this.

4. Change the folder to "Release_0.950", you may be there already.

5. Change to the folder "test-data"; this folder contains the GenBank formatted sequence of phiX174 and a selection of 25,000 short reads in FastQ format. It further contains the needed ini files and the result files generated by R2R.
   Thus you do not actually need to edit the ini-files (except that Unix/Linux and MacOS users need to remove a few lines in the Eval_Reads.ini file before running the R2R_Analyzer.pl).
   Create Genome Index and Annotation files:

6. Change to the folder "Genome". Run the script with the GB formatted sequence as argument:
   ```
   > ..\..\R2R_Annotation_Extraction.pl NC_001422_phiX.gb     [Windows]
   > ../../R2R_Annotation_Extraction.pl NC_001422_phiX.gb     [Lin/Uni/Mac]
   ```
   Note for Mac users: MacOS might not recognize the script files as Perl scripts. In that case try:
   ```
   > perl ../../R2R_Annotation_Extraction.pl NC_001422_phiX.gb
   ```
   for this and the remaining scripts of this tutorial.

7. This should result in two new files:
   NC_001422_phiX.gb.cds and
   NC_001422_phiX.gb.fa
   the .cds contains annotation information to be used much later when assembling panels.
   the .fa file is used now as well as in the next steps.

8. Edit the "Genome_Index.ini" file so it contains these parameters:
   $chr = NC_001422_phiX.gb.fa;
   $chr_end = y
   $name = phiX_34
   $rd_length = 34
   $start_no = 1
   $index_dublicates= y

9. Run the script:
   ```
   > ..\..\R2R_Genome_Index.pl        [Windows]
   > ../../R2R_Genome_Index.pl        [Lin/Uni/Mac]
   ```

   this results in 4 new files:
   phiX_34.discarded_seqs
   phiX_34.GI_report
   phiX_34.srt_seqs
   phiX_34.NC_001422_phiX.gb.fa.repeat.csv

   .discarded_seqs informs you about sequences obviously not included further (sequences near any [^acgt] letters in the sequence - FYI only.
   GI_report is a receipt – FYI only.
   .srt_seqs is the reference sequence index, will be used for mapping reads.
   .repeat.csv contains information on any sequence of length $rd_length = 34 repeated one or more times. The file is empty, since phiX174 does not have such long repeats. To be used much later when assembling panels.

Index reads:

10. Change up to the folder "test-data".
    Edit the "R2R_read.ini" file so it contains these parameters:
    $fid_fq = s_5_sequence;
    $fid_core = phiX_tst1;
    $qbase = 64;
    $qlength = 28;
    $qlimit = 30;

11. Run the script:
    ```
    > ..\R2R_Read_Index.pl        [Windows]
    > ../R2R_Read_Index.pl        [Lin/Uni/Mac]
    ```

    this results in 3 new files:
    phiX_tst1.sort_reads
    phiX_tst1.readno_srt
    phiX_tst1.lq_reads.fq
    phiX_tst1.Read_report.txt

    All names start with the value of $fid_core.
    .sort_reads contains the read index , will be used for mapping reads.
    .readno_srt contains the original reads, will be used to save un-mapped reads.

.lq_reads.fq contains reads filtered away due to either 1) contains n's or other non-acgt bases or 2) quality is below set limit - FYI only.
.Read_report.txt contains statistics and other information - FYI only.

Map the reads to the reference sequence:

12. Edit (again) the "R2R_read.ini" file so it contains these *additional* parameters:

```
$reference_in = Genome\phiX_34.srt_seqs;
$ra_qlength = 30;
$ra_qlimit =  30;
$mut_qlimit = 30;
$mut_max = 2;
$init_treshold = 9;
$unik_treshold = 12;
```

13. Run the script:

```
> ..\R2R_Mapper.pl        [Windows]
> ../R2R_Mapper.pl        [Lin/Uni/Mac]
```

this results in 4 new files:

phiX_tst1.orphans.fq contains orphan reads in fastq format
phiX_tst1.errors.txt may contain error messages
phiX_tst1.Morph_report.txt contains statistics and other information - FYI only.
phiX_tst1.mut_discarded contains information on discarded reads - FYI only.

and in a (new) sub-folder named "NC_001422_phiX.gb.fa.dir":

phiX_tst1_mo_srt.bex
phiX_tst1_rd_srt.bex

containing mapped mutation / indel reads and matching reads respectively. Both .bex files will be used for the final analysis.

Final analysis:

14. Change folder:

```
> cd NC_001422_phiX.gb.fa.dir   [all systems]
```

15. Edit the "Eval_Reads.ini" file so it contains at least these parameters.
(Note: Windows users may or may not need to delete or mask three Unix/Linux/Mac specific lines if you are using the sample file.)

```
$fid_core = phiX_tst1;
$repeat_csv = ../Genome/phiX.NC_001422_phiX.gb.fa.repeat.csv;      # if Linux/Unix
$repeat_csv = ..\Genome\phiX.NC_001422_phiX.gb.fa.repeat.csv;      # if Windows
$fa = ../Genome/NC_001422_phiX.gb.fa;                              # if Linux/Unix
$fa = ..\Genome\NC_001422_phiX.gb.fa;                              # if Windows
$anno = ../Genome/NC_001422_phiX.gb.cds;                           # if Linux/Unix
$anno = ..\Genome\\NC_001422_phiX.gb.cds;                          # if Windows
$qbase = 64;
$start_no = 1;
$print_reads = y;
```

```
$print_read_density =  y;
$print_mut_quality = y;
$panel_width = 1000;
$seq_overlap = 7;
$min_pr = 1;
$ind_min = 4;
$mut_min = 4;
$mut_freq_min = 1;
$ID_limit = 1;
$no_mut_in_repeats =  y;
$check_translation =  y;
$read_dist = 8;
```

16. Run the script:

```
> ..\..\R2R_Analyzer.pl        [Windows]
> ../../R2R_Analyzer.pl        [Lin/Uni/Mac]
```

this results in 5 new files:

phiX_tst1.Index.htm
phiX_tst1.Report_AUX.txt
phiX_tst1.Report_Mutations.htm
phiX_tst1.Report_Reads.csv
phiX_tst1.Report_Unconfirmed_bases.htm

and a sub-folder “html” with “many” .htm files, one for each panel
Each of the .htm files provides index and links to relevant /html/.htm files, open them with your favorite browser.
.Report_AUX.txt, phiX_tst1.Report_Mutations.txt, and phiX_tst1.Report_Reads.csv contain statistics and other information - FYI only.

17. Open phiX_tst1.Report_Mutations.htm with your web-browser. The line:

```
Panel:    0   587  Mutation:     173    122    51     0   ...
```

indicates a mutation, click on the link and the panel should open in a separate window or tab. With IE explorer and Mozilla Firefox you are also directed near position 587. A “g” in the reference sequence is an “a” in all the reads of the sequenced sample (1.00).

Now you have found your first mutation with R2R. This mutation changes the gtg (valine) code to gta (valine) [V>V;] in the D protein and the tgg (tryptophane) to tga (stop) [W>*;] in the E protein.

Use your intuition and the rest of this manual to progress from here.

Appendix F: Ascii codes

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Appendix G: Codon Table

RNA codon table

nonpolar | polar | basic | acidic | (stop codon)

The table shows the 64 codons and the amino acid for each. The direction of the mRNA is 5' to 3'.

		2nd base			
		U	**C**	**A**	**G**
1st base	**U**	UUU (Phe/F) Phenylalanine	UCU (Ser/S) Serine	UAU (Tyr/Y) Tyrosine	UGU (Cys/C) Cysteine
		UUC (Phe/F) Phenylalanine	UCC (Ser/S) Serine	UAC (Tyr/Y) Tyrosine	UGC (Cys/C) Cysteine
		UUA (Leu/L) Leucine	UCA (Ser/S) Serine	UAA Ochre (*Stop*)	UGA Opal (*Stop*)
		UUG (Leu/L) Leucine	UCG (Ser/S) Serine	UAG Amber (*Stop*)	UGG (Trp/W) Tryptophan
	C	CUU (Leu/L) Leucine	CCU (Pro/P) Proline	CAU (His/H) Histidine	CGU (Arg/R) Arginine
		CUC (Leu/L) Leucine	CCC (Pro/P) Proline	CAC (His/H) Histidine	CGC (Arg/R) Arginine
		CUA (Leu/L) Leucine	CCA (Pro/P) Proline	CAA (Gln/Q) Glutamine	CGA (Arg/R) Arginine
		CUG (Leu/L) Leucine	CCG (Pro/P) Proline	CAG (Gln/Q) Glutamine	CGG (Arg/R) Arginine
	A	AUU (Ile/I) Isoleucine	ACU (Thr/T) Threonine	AAU (Asn/N) Asparagine	AGU (Ser/S) Serine
		AUC (Ile/I) Isoleucine	ACC (Thr/T) Threonine	AAC (Asn/N) Asparagine	AGC (Ser/S) Serine
		AUA (Ile/I) Isoleucine	ACA (Thr/T) Threonine	AAA (Lys/K) Lysine	AGA (Arg/R) Arginine
		AUG[A] (Met/M) Methionine	ACG (Thr/T) Threonine	AAG (Lys/K) Lysine	AGG (Arg/R) Arginine
	G	GUU (Val/V) Valine	GCU (Ala/A) Alanine	GAU (Asp/D) Aspartic acid	GGU (Gly/G) Glycine
		GUC (Val/V) Valine	GCC (Ala/A) Alanine	GAC (Asp/D) Aspartic acid	GGC (Gly/G) Glycine
		GUA (Val/V) Valine	GCA (Ala/A) Alanine	GAA (Glu/E) Glutamic acid	GGA (Gly/G) Glycine
		GUG (Val/V) Valine	GCG (Ala/A) Alanine	GAG (Glu/E) Glutamic acid	GGG (Gly/G) Glycine

[A] The codon AUG both codes for methionine and serves as an initiation site: the first AUG in an mRNA's coding region is where translation into protein begins.

From: http://en.wikipedia.org/wiki/Genetic_code