Reference	SeqQual pipeline / program (prog) name*,¤	usage**	arguments (arg)	description	using programs (other prog)	Shell examples **	Input file(s)	Output folder/file(s)	log file available$
Files management*									
Lang et al.	checkdir_mydata.pl	perl prog		create new folder mydata, if one already exists--> change it to mydata_current_date&time first				empty folder mydata	
"	checkdir_Output.pl	perl prog		create new folder Output, if it already exists, change it to Output_current_date&time first		*fasta-*data.sh		empty folder Output	
"	checkinput.pl	perl prog arg	text file with list of folder names, e.g. inputile	check if the folders listed in inputfile exist, give a warning if not				screen	
"	**print_source**-take_aln.pl¤	**(perl prog arg > out source out) =** ***(A)***	(text file with list of folder names, e.g. inputile) = ***(B)***	(run **other prog** in batch on a set of folder given as arg , when sourcing the "out" text file) = ***(C)***	take_aln_to_output.pl	*fasta-*data.sh			
"	take_aln_to_output.pl	perl prog arg	***(B)***	look for files with extension ".aln"; rename as filenamenumber.aln, take file to Output/aln folder				needs to be created, see example *.sh for creation of Output/aln folder	
Fasta alignments editing and filtering for subsequent analyses									
Lang et al.	merge_multinput.pl	perl prog arg	number of characters at start of read/sequence name	merge any sequences/reads which id/name starts by the same arg (number of characters), any mismatch is replaced by N in the merged sequence. Exceptions if nucleotides are present with N or ?, nucleotides are given		mistools.sh	*.aln fasta files	aln_/log_merged / *.aln fasta files	**X**
"	merge_multinput_IUPAC.pl	perl prog arg	number of characters at start of read/sequence name	merge any sequences/reads which id/name starts by the same arg number of characters. 2 nucleotides are replaced by IUPAC codes, any other mismatch is replaced by N in the merged sequence, except for nucleotides with N or ? for which nucleotides are given		"	*.aln fasta files	aln_/log_merged / *.aln fasta files	**X**
"	**merge_multinput_F_R.pl**	perl prog		merge reads with same id & any suffix like "_01/2/3/4/5/6etc-P30-F/R", adjusted to our data output with Phred 30 from SeqQual, any mismatch is replaced by N		"	*.aln fasta files	aln_/log_merged / *.aln fasta files	**X**
"	split_aln_multi.pl	perl prog arg	nb of bp after which each file is split	looks for all *.aln/*.fas files in same folder (add folder in path otherwise), will create split files in a new folder according to arg and rename them with corresponding suffix,			*.aln/*.fas fasta files	aln_splitted/*.aln fasta files	
El mujtar et al. 2014	delete_empty_seqs.pl	perl prog > log.txt		Looks for *.aln/*.fas files. Delete read lines which only contain "???" Or "---"(empty seqs, for ex after masking steps) and produces new files without those lines in a new folder. Names them "OriginalSuffix".del.aln/fas.			*.aln fasta files	aln_del_empty_seq/*.aln fasta files	
Brousseau et al. 2014, El mujtar et al. 2014	**get_first.pl**	perl prog		Looks for *.aln files. Produces as many files as fasta alignments only including the first line (assumed to be the consensus) in a new folder. Names them "OriginalSuffix_consensus.aln". Also gives all first lines in one fasta file.		"	*.aln fasta files	aln_consensus/*.aln fasta files	
El mujtar et al. 2014	homomaskAln.pl	perl prog arg > log.txt	nb of nucleotides before or after the insertion for which the deletions/insertions are masked, works also for SNPs with false deletions in those homopolymer regions	Looks for *.aln files. Produces files in which some deletions are masked (replaced by "?") in a new folder. The deletions to be masked occur in homopolymers regions for a defined number of repeated nucleotides (the arg). Names the files "OriginalSuffix.homomasked.aln".			*.aln fasta files	aln_homomask/*.aln fasta files	
Brousseau et al. 2014, El mujtar et al. 2014	**make_consensus_maxallele_N**.pl§	perl prog		Look for *.aln files (assumed haploid data), compute in each consensus sequence using the allele in highest frequency, and add it as first sequence, move alignments in a new folder, name them "**OriginalSuffix.nc**.aln".			***.aln fasta files for haploid reads/sequences**	**aln_cons_maxal/***.aln fasta files	
Lang et al.	make_consensus_IUPAC_2N/maxallele_2N	perl prog		Look for *.aln files, compute consensus sequence (either using IUPAC codes when there is a SNP, ignoring deletions if present (for make_consensus_IUPAC_2N.pl) or using the allele in highest frequency (make_consensus_maxallele_2N.pl) and add it as first sequence in the alignment, move the alignments in a new folder, name them "**OriginalSuffix.nc**.aln".	suggest to use remove_first.pl if a consensus is already present as first sequence		***.aln fasta files for diploid sequences**	**aln_cons_IUPAC/***.aln fasta files	
Brousseau et al. 2014	maskAln.pl	perl prog --man --man_if_depth --maf --indel_only	maf= min allele frequency, man=min allele number, man_if_depth=min depth for which masking is on for the man given, indel_only=if yes only performs masking on indels, if no does it across all variants	according to either maf or man, will mask (=replace by "?") any nucleotide which is a variant below the given maf or man.			*.aln fasta files	**aln_mask**/*.aln fasta files	

"	pick-seq.pl	perl prog arg *.aln	optional suite of characters which identify starts of reads names. E.g. "pop1 pop2"	looks for set of *.aln fasta files. Select in each of them a number of reads identified by the arg and produces another set of *.aln files in a separate folder containing those reads			*.aln fasta files	aln_pick/*.aln fasta files	
lang et al.	**print_source**-fasta.pl¤	***(A)***	***(B)***	***(*** run **other prog** in batch on a set of folder given as arg , when sourcing the "out" text file) = ***(C)*** , create working directories and files	checkdir_mydata.pl	*fasta-*data.sh	see arg	mydata/aln_final	**X**
"	**print_source**-replace.pl¤	***(A)***	***(B),*** number (1, 2 or 3) of isolated nucleotides replaced	***(C)*** , dependent on running print_source-fasta.pl first, put result files into folder aln_final	replace-X-nucleotide_aln.pl	"	see arg		**X**
"	replace-1-nucleotide_aln.pl¤¤	perl prog		look for files with extension ".aln"; replace single isolated nucleotides surrounded by at least 5 "?" with "?"		"	*.aln fasta files	aln_replace1/*.aln fasta files	
"	replace-2-nucleotide_aln.pl	perl prog		look for files with extension ".aln"; replace single or 2 neighbour isolated nucleotides surrounded by at least 5 "?" with "?"		"	*.aln fasta files	aln_replace2/*.aln fasta files	
"	replace-3-nucleotide_aln.pl	perl prog		look for files with extension ".aln"; replace single, 2 or 3 neighbour isolated nucleotides surrounded by at least 5 "?" with "?"		"	*.aln fasta files	aln_replace3/*.aln fasta files	
"	**print_source**-truncate.pl¤	***(A)***	***(B),*** threshold number indicating until which position to truncate at start & end	***(C)*** , dependent on running print_source-fasta.pl first, put result files into folder aln_final	trunc_aln.pl	"	see arg		**X**
"	trunc_aln.pl¤¤	perl prog arg	threshold number indicating until which position to truncate at start & end	look for file with extensions ".aln"; truncate start and end of alignment until the positions have more non-missing bases than the arg (e.g. "9" means that start & end will truncated until the first position that has got 10 non-missing nucleotides)			*.aln fasta files	folder aln_trunc with files	
Brousseau et al. 2014	**print_source-remove1.pl**	***(A)***	***(B)***	***(C) ,*** dependent on running print_source-fasta.pl first, put result files into folder aln_final	remove1-pos_aln.pl		see arg	text file to source	**X**
"	**remove1-pos_aln.pl**	perl prog		look for files with extension ".aln"; remove positions (columns) with only "?" and "-", and "-" in consensus (**needs consensus** as first sequence, use **make_consensus*.pl** if not)			*.aln fasta files	aln_remove/*.aln fasta files	
Brousseau et al. 2014, El mujtar et al. 2014	**remove_first.pl**	perl prog		look for *.aln files, produce files without the first line (assumed to be the consensus) in a new folder, name them "OriginalSuffix.removeconsensus.aln".		mistools.sh	*.aln fasta files	**aln_remove_first**/*.aln fasta files	
Lang et al.	**print_source**-write_SNP-fasta_aln-no_first.pl	***(A)***	***(B)***	***(C)*** , dependent on running print_source-fasta.pl first	fasta2snp_no_first.pl	*fasta-*data.sh	see arg		
"	fasta2snp_no_first.pl	perl prog		pick SNP alignment files from fasta files, not taking the first sequence assumed to be the consensus			*.aln fasta files	needs to be created, see example *.sh for creation of Output/aln_haplotype folder	
"	**print_source**-write_SNP-fasta_aln.pl	***(A)***	***(B)***	***(C)*** , dependent on running print_source-fasta.pl first	fasta2snp.pl	*fasta-*data.sh	see arg		
"	fasta2snp.pl	perl prog		pick SNP alignment files from fasta files			*.aln fasta files	needs to be created, see example *.sh for creation of Output/SNP folder	
"	**print_source**-take_SNP-fasta2snp.pl	***(A)***	***(B)***	***(C)*** , dependent on running print_source-fasta.pl first, see *.sh for creating SNP folder	take_SNP-fasta2snp_to_output.pl	*fasta-*data.sh	see arg		
"	take_SNP-fasta2snp_to_output.pl	perl prog		look for file with name filename ".aln.snp"; rename as filename.snp.aln, take file to Output/SNP folder			*.aln.snp fasta files		
"	**print_source**-write_haplotypealn_nofirst.pl	***(A)***	***(B)***	***(C)*** , dependent on running print_source-fasta.pl first	write_haplotype_phase_unknown_multinput_nofirst.pl	*fasta-*data.sh	see arg		
"	write_haplotype_phase_unknown_multinput_nofirst.pl	perl prog		look for files with extension ".aln"; , write phase unknown fasta alignment from diploid alignment without consensus sequence (assumed to be the first line)			*.aln fasta files		
"	**print_source**-write_haplotypealn.pl	***(A)***	***(B)***	***(C)*** , dependent on running print_source-fasta.pl first	write_haplotype_phase_unknown_multinput.pl	*fasta-*data.sh	see arg		
"	write_haplotype_phase_unknown_multinput.pl	perl prog		look for files with extension ".aln"; , write phase unknown fasta alignment from diploid alignment			*.aln fasta files		
"	**print_source**-take_haplotypealn.pl	***(A)***	***(B)***	***(C)*** , dependent on running print_source-fasta.pl first	take_haplotypealn_to_output.pl	*fasta-*data.sh	see arg		
"	take_haplotypealn_to_output.pl	perl prog		look for files with extension ".haplotype.aln"; rename as filenamenumber.haplotype.aln, take file to Output/aln_haplotype folder			*.haplotype.aln fasta files	needs to be created, see example *.sh for creation of Output/aln_haplotype folder	
"	**print_source**-write_unaln-fasta.pl	***(A)***	***(B)***	***(C)*** , dependent on running print_source-fasta.pl first	write_unaln.pl		see arg		
"	write_unaln.pl	perl prog		look for files with extension ".aln"; write unaligned sequences from fasta files to unaln folder (so they can be re-aligned with another/better software if needed)			*.aln fasta files	unaln/*.aln fasta files	
"	**print_source**-take_unaln.pl	***(A)***	***(B)***	***(C)*** , dependent on running print_source-fasta.pl first	take_unaln_to_output.pl		see arg		

"	take_unaln_to_output.pl	perl prog		look for files with extension ".unaln"; take files to Output/unaln folder			*.unaln fasta files		

Fasta alignments post-treatment for genetic analyses and SNP detection**

Brousseau et al. 2014, El mujtar et al. 2014	**print_source**-SNP_statistic_haplo.pl¤	*(A)*	*(B)* ; (optional set of characters identifying start of sequence/read names and corresponding to up to 2 different groups, for ex: "g1 g2"; if no arg is given, all reads are treated as a single group) = *(D)*	*(C)* , works for haploid sequences/reads, **see example *.sh file**	SNP-statistic0/1/2-haplo.pl	3.3-SNP_statistic-diplo-haplo.sh	see arg		
"	SNP_statistic0/1/2-haplo.pl¤¤	perl prog arg > result.txt	*(D)*	computes counts of different polymorphisms, depth, min allele frequency, and more if 2 groups (exclusive & shared alleles, divergence statistics (Gst, Gst'), etc..., see result.txt file)		"	*.aln fasta files	tabulated SNP_statistic*.txt in the folder in which prog is called	
lang et al.	**print_source**-SNP_statistic.pl	*(A)*	*(B)* ; *(D)*	*(C)* , works for diploid sequences, **see example *.sh file**	SNP-statistic0/1/2.pl	"	see arg		
"	SNP_statistic0/1/2.pl	perl prog arg > result.txt	*(D)*	computes counts of different polymorphisms, depth, min allele frequency, HW chi2 test, and more if 2 groups (exclusive & shared alleles, divergence statistics (Gst, Gst'), etc...)		"	*.aln fasta files	tabulated resultfile.txt (default name) as the program runs in the same folder	
"	**print_source**-arlequin-diploid.pl	*(A)*	*(B)* ; (optional set of characters identifying start of sequence/read names and corresponding to up to 20 different groups, for ex: "g1 g2 g3"; if no arg is given, all reads are treated as a single group, **can easily be changed by the user) = *(E)***	*(C)* , put arlequin formatted (*.arp) files into folder arlequin_input0 (for format with haplotype phase unknown) and arlequin_input1 (for format with genotypic diploid sequences with IUPAC codes)	write_arlequin_input_diploid-genotypicdata0_multinput.pl, write_arlequin_input_diploid-genotypicdata1_multinput.pl	*fasta-*data.sh	see arg		**X**
"	write_arlequin_input_diploid-genotypicdata0_multinput.pl		*(E)*	look for fasta files with extension ".aln"; write "GenotypicData=0" arlequin input file from diploid data in fasta alignments, producing haplotype sequence phase unknown (but note that Arlequin considers that the phase is known in this format, so summary statistics are only correct if they are not affected by haplotye phase. Also produces the arb files (list of arp files) for batch analyses in Arlequin			*.aln fasta files	folder arlequin_input0 with files	
"	write_arlequin_input_diploid-genotypicdata1_multinput.pl		*(E)*	look for fasta files with extension ".aln"; write "GenotypicData=1" and "GameticPhase=0" (phase unknown ackowledged) Arlequin input files from diploid data in fasta alignments (*.dip.arp that can be used for phasing haplotypes in Arlequin)			*.aln fasta files	folder arlequin_input1 with files	
"	**print_source**-arlequin-haploid.pl	*(A)*	*(B)* ; *(E)*	*(C)* , put arlequin formatted (*.arp) files into folder arlequin_input	write_arlequin_input_multinput.pl	*fasta-*data.sh	see arg		**X**
"	write_arlequin_input_multinput.pl		*(E)*	look for fasta files with extension ".aln"; write haplotype sequence arlequin input file from haploid data fasta alignments			*.aln fasta files	folder arlequin_input with files	
"	**print_source**-take_arp_diploid.pl	*(A)*	*(B)*	*(C)* , take diploid arlequin input files (two types) to Output folder, also creates the arb file (list of arp files) for batch analyses in Arlequin	take_arp_to_output_diploid.pl	*fasta-*data.sh	see arg		
"	take_arp_to_output_diploid.pl		file with list of folder names	look for files with extension ".arp"; rename as filenamenamenumber.dip(/hap).arp (depending on whether filename contains "genotypicdata0" or "genotypic data1"), take file to Output/arlequin folder			*.aln fasta files		
"	**print_source**-take_arp_haploid.pl	*(A)*	*(B)*	*(C)* , take arlequin input files from haploid data to Output folder, also creates the arb file (list of arp files) for batch analyses in Arlequin	take_arp_to_output_haploid.pl	*fasta-*data.sh	see arg		
"	take_arp_to_output_haploid.pl		file with list of folder names	look for file with name filename ".arp"; rename as filename.aln.arp, take file to Output/arlequin folder			*.aln fasta files		

* all shell scripts can be run by typing "source *.sh"

¤ **All print_source scripts work by printing a txt file that needs to be sourced to launch other programs for batch treatment of files located in one or different folders. They also require a particular folder structure for printing results files (see start of example *.sh files for details)**

¤¤ **But all other programs can be used also independently**

** To run print_source*.pl prog, most other programs are assumed to be located under home/SeqQual but this can easily be changed in the code

$ see log related scripts in **SeqQual_log.pdf**

*** these programs are needed if the print_source programs are used

§ these programs have been modified/or bug corrected compared to original publications

in blue: script's names changed compared to original publications