Appendix 1: Overview of the genetic summary statistics available in the current version of `skeleSim`. Analyses are nested within the categories ‘Locus’, ‘Global’, and ‘Pairwise’. `skeleSim` accesses functions within existing `R` packages (see footnotes for packages used and corresponding object type).

Analysis Type	Locus Type*	**`SkeleSim`** notation	Description	Object and Ref	R function: what it looks like in R	Function explanation given by R package
Global	MS/ Seq	`Chi2`	X^2 test statistic for pop struct	`gtypes`[1]	`overallTest(g, stats="chi2", nrep=100)`	Population structure statistic
Global	MS/ Seq	`Chi2.pval`	*p*-value for X^2	`gtypes`[1]	derived from function above (nrep)	*p*-value for population structure statistic
Global	MS/ Seq	`Fst`	F_{ST}, F-statistic	`gtypes`[1]	`overallTest(g, stats="fst", nrep=100)`	Population structure statistic
Global	MS/ Seq	`Fst.pval`	*p*-value for F_{ST}	`gtypes`[1]	derived from function above (nrep)	*p*-value for population structure statistic
Global	MS/ Seq	`F'st`	F'_{ST}	`gtypes`[1]	`overallTest(g, stats="fst.prime", nrep=100)`	Population structure statistic
Global	MS/ Seq	`F'st.pval`	*p*-value for F'_{ST}	`gtypes`[1]	derived from function above (nrep)	*p*-value for population structure statistic
Global	MS	`Fis`	F_{IS}	`gtypes`[1]	`overallTest(g, stats="fis", nrep=100)`	Population structure statistic
Global	MS	`Fis.pval`	*p*-value for F_{IS}	`gtypes`[1]	derived from function above (nrep)	*p*-value for population structure statistic
Global	MS	`Gst`	G_{ST}	`gtypes`[1]	`overallTest(g, stats="gst", nrep=100)`	Population structure statistic
Global	MS	`Gst.pval`	*p*-value for G_{ST}	`gtypes`[1]	derived from function above (nrep)	*p*-value for population structure statistic
Global	MS	`G'st`	G'_{ST}	`gtypes`[1]	`overallTest(g, stats="gst.prime", nrep=100)`	Population structure statistics. Type of G'_{ST} to calculate can be "nei" or "hedrick"
Global	MS	`G'st.pval`	*p*-value for, G'_{ST}	`gtypes`[1]	derived from function above (nrep)	*p*-value for population structure statistic
Global	MS	`G''st`	G''_{ST}	`gtypes`[1]	`overallTest(g, stats="gst.dbl.prime",`	Population structure statistic

					`nrep=100)`	
Global	MS	`G''st.pval`	*p*-value for G''_{ST}	`gtypes`[1]	derived from function above (nrep)	*p*-value for population structure statistic
Global	Seq	`Phist`	Φ-statistic	`gtypes`[1]	`overallTest(g, stats="phist", nrep=100)`	Population structure statistic
Global	Seq	`Phist.p.val`	*p*-value for Φ	`gtypes`[1]	derived from function above (nrep)	*p*-value for population structure statistic
Locus	MS	`allelic.richness`	Allelic richness	`gtypes`[1]	`allelicRichness(g)`	The allelic richness of each locus calculated as num. alleles divided by num. of samples without missing data at that locus
Locus	MS	`num.alleles`	Number of alleles	`gtypes`[1]	`numAlleles(g)`	Return the number of alleles for each locus
Locus	MS	`prop.genotyped`	Proportion genotyped	`gtypes`[1]	`prop.genotyped(g)`	Delivers standard by-locus summaries, including proportion of samples genotyped
Locus	MS	`theta`	θ, theta	`gtypes`[1]	`theta(g)`	Calculate θ from heterozygosity of each locus
Locus	MS	`exptd.heterozygosity`	Expected heterozygosity	`gtypes`[1]	`exptdHet(g)`	Expected heterozygosity for diploid data
Locus	MS	`obsvd.heterozygosity`	Observed heterozygosity	`gtypes`[1]	`obsvdHet(g)`	Observed heterozygosity for diploid data
Locus	MS	`Fis`	F_{IS}, Inbreeding coefficient	`loci`[2]	`Fst(loci)`	A matrix with loci as rows and three F-statistics as columns. Formulae are used per-allele, then averaged within each locus over the alleles[4].
Locus	MS	`Fit`	F_{IT}, Overall fixation index	`loci`[2]	`Fst(loci)`	A matrix with loci as rows and three F-statistics as columns. Formulae are used per-allele, then averaged within each locus over the alleles[4].
Locus	MS	`Fst`	F_{ST}, Fixation index	`loci`[2]	`Fst(loci)`	A matrix with loci as rows and three F-statistics as columns. Formulae are used per-allele, then averaged within each locus over the alleles[4].

Locus	MS	`hwe.p`	Hardy-Weinberg equilibrium	`gtypes`[1]	`hweTest(g, use.genepop=FALSE, B=1000, label="HWE.genepop", ...)`	Calculate Hardy-Weinberg equilibrium p-values.
Locus	MS	`num.priv.allele`	Number of private alleles	`gtypes`[1]	`privateAlleles(g)`	The num. private alleles in each strata and locus
Locus	MS	`prop.unique.alleles`	Proportion of unique alleles	`gtypes`[1]	`propUniqueAlleles(g)`	Proportion of unique (occurring in one individual) alleles for each locus
Locus	MS	`mRatio`	M-ratio, bottleneck statistic	`gtypes`[1]	`mRatio(g)`	Calculate M-ratio (bottleneck) statistic for microsatellite data[5]
Locus	Seq	`hap.richness`	Haplotype richness	`gtypes`[1]	`allelicRichness(g)`	The haplotype richness of each locus calculated as num. haplotypes divided by num. samples without missing data at that locus
Locus	Seq	`num.haps`	Number of haplotypes	`gtypes`[1]	`numAlleles(g)`	Return the number of haplotypes for each locus
Locus	Seq	`nucleotide.diversity`	π, nucleotide diversity	`gtypes`[1]	`nucleotideDiversity(g)`	Calculate nucleotide diversity for set of haplotypes (nucleotide diversity by site).
Locus	Seq	`nucleotide.divergence`	dA, mean nucleotide divergence within	`gtypes`[1]	`dA <- nucleotideDivergence(g); dA[[1]]$within`	Calculate Nei's dA between strata, and distributions of within-strata nucleotide divergence (seq. distance)[6].
Locus	Seq	`Heterozygosity`	Heterozygosity/ haplotype diversity	`gtypes`[1]	`exptdHet(g)`	A measure of haplotypic diversity (haploid "heterozygosity")
Locus	Seq	`num.private.haps`	Number of private haplotypes	`gtypes`[1]	`privateAlleles(g)`	The number of private haplotypes in each strata and locus.
Locus	Seq	`pct.unique.haps`	Percent unique haplotypes	`gtypes`[1]	`propUniqueAlleles(g)`	The proportion of unique haplotypes (occurring only in one individual).
Locus	Seq	`Fu.F`	Fu's F_S	`gtypes`[1]	`FusFs(g)`	Calculate Fu's F_S for a set of sequences to test for selection[7].
Locus	Seq	`Fu.F.p.val`	p-value for Fu's F_S	`gtypes`[1]	`derived from function above`	p-value for Fu's F_S
Locus	Seq	`Taj.D`	Tajma's D statistic	`gtypes`[1]	`tajimasD(g)`	Calculate Tajima's D for a set of sequences to test for selection[8].

Locus	Seq	`Taj.D.p.val`	*p*-value for Tajma's *D*	`gtypes`[1]	`derived from function above`	*p*-value for Tajima's *D*
Pairwise	MS/ Seq	`Chi2`	X^2 test statistic for pairwise population differentiation	`gtypes`[1]	`pairwiseTest(g, nrep=1000, stats="chi2", ...)`	Pairwise test of genetic differentiation
Pairwise	MS/ Seq	`Chi2.p.val`	*p*-value for X^2 test for pairwise population differentiation	`gtypes`[1]	`derived from function above`	*p*-value for pairwise test of genetic differentiation
Pairwise	MS/ Seq	`Fst`	pairwise F_{ST}	`gtypes`[1]	`pairwiseTest(g, nrep=1000, stats="fst", ...)`	Pairwise test of genetic differentiation
Pairwise	MS/ Seq	`Fst.p.val`	p-value for pairwise F_{ST}	`gtypes`[1]	`derived from function above`	*p*-value for pairwise test of genetic differentiation
Pairwise	MS	`F'st`	pairwise F'_{ST}	`gtypes`[1]	`pairwiseTest(g, nrep=1000, stats="fst.prime", ...)`	Pairwise test of genetic differentiation
Pairwise	MS	`F'st.p.val`	p-value for pairwise F'_{ST}	`gtypes`[1]	`derived from function above`	*p*-value for pairwise test of genetic differentiation
Pairwise	MS	`Fis`	pairwise F_{IS}	`gtypes`[1]	`pairwiseTest(g, nrep=1000, stats="fis", ...)`	Pairwise test of genetic differentiation
Pairwise	MS	`Fis.p.val`	p-value for pairwise F_{IS}	`gtypes`[1]	`derived from function above`	*p*-value for pairwise test of genetic differentiation
Pairwise	MS	`Gst`	pairwise G_{ST}	`gtypes`[1]	`pairwiseTest(g, nrep=1000, stats="gst", ...)`	Pairwise test of genetic differentiation
Pairwise	MS	`Gst.p.val`	p-value for pairwise G_{ST}	`gtypes`[1]	`derived from function above`	*p*-value for pairwise test of genetic differentiation
Pairwise	MS	`G'st`	pairwise G'_{ST}	`gtypes`[1]	`pairwiseTest(g, nrep=1000, stats="gst.prime", ...)`	Pairwise test of genetic differentiation
Pairwise	MS	`G'st.p.val`	*p*-value for pairwise G'_{ST}	`gtypes`[1]	derived from function above	*p*-value for pairwise test of genetic differentiation
Pairwise	MS	`G''st`	pairwise G''_{ST}	`gtypes`[1]	`pairwiseTest(g,`	Pairwise test of genetic differentiation

					`nrep=1000, stats="gst.dbl.prime", ...)`	
	MS	`G''st.p.val`	*p*-value for pairwise G''_{ST}	`gtypes`[1]	derived from function above	*p*-value for pairwise test of genetic differentiation
Pairwise	MS/ Seq	`chord.dist`	Chord distance	`dat`[3]	`genet.dist(dat, diploid=TRUE, method="Dch")`	Chord distance (eqn #6 in[9]) is returned. This distance is used as default since it is best to retrieve the relation among samples[10].
Pairwise	MS/ Seq	`shared.alleles`	Number of shared alleles or haplotypes	`gtypes`[1]	`sharedAlleles(g)`	The number alleles shared among populations
Pairwise	Seq	`nucleotide.divergence`	Mean nucleotide divergence between	`gtypes`[1]	`dA <- nucleotideDivergence(g); dA[[1]]$between`	Calculate Nei's dA between strata, and distributions of between-strata nucleotide divergence (seq distance)[6].
Pairwise	Seq	`Phist`	Φ-statistic	`gtypes`[1]	`pairwiseTest(g, nrep=1000, stats="phist", ...)`	Pairwise test of genetic differentiation
Pairwise	Seq	`Phist.p.val`	*p*-value for Φ	`gtypes`[1]	derived from function above	*p*-value for pairwise test of genetic differentiation

*Locus type is either microsatellite (MS), sequence (Seq), or both (MS/Seq).

[1] `g`, a `gtypes` object; Archer FI, Adams PE, Schneiders B. strataG: An R package for manipulating, summarizing, and analyzing population genetic data. *In submission, Molecular Ecology Resources.*

[2] a `loci` object; Paradis E (2010) pegas: an R package for population genetics with an integrated-modular approach. *Bioinformatics* 26: 419-420.

[3] `dat`, a `data.frame`; Goudet J (2005) hierfstat, a package for R to compute and test hierarchical F-statistics. *Molecular Ecology Notes* 5: 184–186.

[4] Weir BS, Cockerham CC (1984) Estimating F-statistics for the analysis of population structure. *Evolution* 38: 1358–1370.

[5] Garza JC, Williamson EG (2001) Detection of reduction in population size using data from microsatellite loci. *Molecular Ecology* 10: 305-318.

[6] Nei M, Kumar S (2000) *Molecular Evolution and Phylogenetics.* Oxford University Press, Oxford. (dA: pp. 256, eqn 12.67)

[7] Fu Y-X (1997) Statistical tests of neutrality of mutations against population growth, hitchiking and background selection. *Genetics* 147:915-925.

[8] Tajima F (1989) Statistical method for testing the neutral mutation hypothesis by DNA polymorphism. *Genetics* 123:585-595.

[9] Cavalli-Sforza LL, Edwards AW (1967) Phylogenetic analysis. Models and estimation procedures. *American Journal of Human Genetics* 19 (3 Pt 1): 233.

[10] Takezaki N, Nei M (1996) Genetic distances and reconstruction of Phylogenetic trees from microsatellite DNA. *Genetics* 144:389-399.