‘GUGS’
General Utilities for Genotyping Study
User’s manual

2021 Oct.
ver. 1.01

Tokurou Shimizu

Introduction

General Utilities for Genotyping Study (GUGS) is a VBA application for MS Excel. GUGS extends more than 100 functionalities for MS Excel to allow genotype data format conversion for linkage analysis to integrate with MapMaker or JoinMap, frequency analysis, parentage analysis by a statistical genetic approach, and nucleic acid or amino acid sequence manipulation. Three major genotype formats, simple sequence repeat (SSR, aka short tandem repeat; STR), single nucleotide polymorphism (SNP) and single-letter genotype are supported with their allele in GUGS. All of these capabilities of GUGS are implemented as a function, and users can combine them with the built-in functions of MS Excel seamlessly. Users are not requested to sort or format the data before using GUGS; users can execute all the functions at any place in the spreadsheet.

The current version of GUGS does not support an advanced analysis, such as simulation-based methods (e.g., MCMC), phylogenetic analysis (UPGMA, Neighbor-Joining, maximum likelihood, or else), or linkage map construction because various excellent applications are already available. GUGS evaluates genotype data and converts their formats for further analysis in a single environment without any efforts to export and import data. Users can confirm the usage of individual functions by referring to the ‘GUGS workbook’ which is included in the GUGS.xlsm file.

GUGS is distributed under a GNU General Public License version 3 (GPL3) license.

21th Oct. 2017

Tokurou Shimizu

Table of Contents

1. Environments

GUGS is confirmed to work with MS Excel 2010, 2013, or 2016 including Office 365 on MS Windows 7 or later, or Mac OS.

2. Installation and execution

General Utilities for Genotyping Study, GUGS, is a VBA application of MS Excel. It is distributed as an Excel file implemented with VBA (GUGS.xlsm). Executable GUGS with MS Excel is available at https://github.com/tokurou/GUGS. No prerequisite step is required for installation or launching GUGS. However, MS Excel prohibits the automatic VBA execution in default; therefore, users need to enable VBA execution asked when launching GUGS. Any user data should be saved in 'Excel Macro-Enabled Workbook (*.xlsm)' format to keep GUGS functionalities.

The performance of GUGS depends on the system memory, clock speed, and the number of cores of the CPU. As an example, running GUGS with 4 cores/8 threads CPU on a Core i7 3.4 GHz with 16 GB memory on a 64-bit MS Windows 10 environment will complete data format conversion for 10,000 SNP genotype to the CP mode of JoinMap within a few seconds.

3. Security issues

GUGS has been confirmed to be free of malware or viral code using several antivirus checkers and with Virus Total (https://www.virustotal.com/en/). However, it is strictly recommended to pay much attention when using this with other VBA code.

4. Summary of functions

GUGS extends more than 100 functions for genotype data manipulation and nucleic acid/amino acid sequence in MS Excel. These functions are classified into the following six groups.

1. Data format conversion

Functions in this category allow to normalize genotype data, convert the absolute size to the relative size of an SSR genotype, insert a separator into an SNP genotype, convert a single-letter genotype into the pseudo-SNP format and extract the difference between two alleles of SSR genotype. It also allows converting Illumina's single-letter genotype format into an SNP genotype, and a converting single-letter HapMap code into a real SNP genotype.

2. Basic analysis

Functions in this category allow to extract allele of an SSR or SNP genotype, examine homozygosity, determine the identity between two genotypes of SSR or SNP format, examine allele inclusion in the given genotype, evaluate the ploidy of an SSR genotype, separate all alleles of SSR or SNP genotypes and extract the shared allele between two SSR or SNP genotypes.

3. Linkage analysis

Functions in this category involve to estimate a segregation mode of offspring from the genotype data of parents and convert the genotype of offspring for evaluation with linkage analysis software such as MapMaker (Lander *et al.*, 1987), JoinMap (Stam, 1993), or their equivalent software.

4. Dataset analysis

Functions in this category allow to estimate unique alleles or genotypes in the dataset, extract an allele or genotype at a designated rank in the dataset, and count the ratio of the matched allele or genotype in the dataset.

5. Frequency analysis

Functions in this category allow estimating allele frequencies or diversity of population data, observed or expected heterozygosity, polymorphic information content (PIC), match probability, the power of discrimination, genetic diversity, and the probability of showing an identical genotype for different individuals in a population dataset according to the theorem of Ukai.

6. Genetic data analysis

Functions in this category are valid to estimate parentage of offspring within a population and determine the probability to obtain an individual from a cross of designated parents in a population.

7. Sequence manipulation

Functions in this category allow for converting, counting, formatting and searching for nucleotide or amino acid sequence motif to support DNA marker design.

5. Data types

GUGS accepts SSR and SNP genotype with their alleles, three types of single-letter genotypes, and related data formats for input, output, or for both.

- **SSR genotype** format is two integers combined with a separator (a slash is used as the separator). A separator is mandatory for the SSR genotype. A single integer value is recognized as an allele of SSR. Any SSR genotype that consists of multiple sets of integers for polyploid data is acceptable, but the current version of GUGS assumes all SSR genotypes as diploid except for the ‘SSRPloidy’ and ‘SSRAllele’ functions.

- **SNP genotype** format consists of two alphabetic letters. A single alphabetic letter is recognized as an allele of SNP. The separator is optional, and it will be ignored. Any length of characters is acceptable, but the initial two letters are valid for the evaluation except for the ‘SNPAllele’ function.

- **Single-letter genotype** is represented as an alphabetic letter. GUGS supports three different types of single-letter genotypes. The first is the simplified nucleotide that has been used in a HapMap project (referred to as ‘HMP’ hereafter). HMP represents an ambiguous nucleotide according to the IUPAC nucleotide notation (https://en.wikipedia.org/wiki/Nucleic_acid_notation). The second is a single-letter genotype for linkage analysis (referred to ‘M’ hereafter). This is a standard genotype format in MapMaker, JoinMap, or similar software (‘A’, ‘B’, or ‘H’). These genotypes represent the homozygote that is identical to the parent A (‘A’), another homozygote that is identical to the parent B (‘B’), and the heterozygote of these parents A and B (‘H’). The third type resembles the second type (‘A’, ‘B’, or ‘H’), but it corresponds to the abstracted SNP that represent the homozygote of the major allele (‘A’), the homozygote of the minor allele (‘B’), or the heterozygote of both alleles (‘H’) (referred to ‘CODE’ hereafter). Most of the SNP analysis software, such as Illumina’s Genome Studio ®, export the third type as output or accept it as input.

- **Other data formats**:

  - **Numeric**: the difference between two SSR alleles or the ploidy of an SSR genotype.
  - **CP**: a particular set of a two-letter genotype used for the CP mode of JoinMap.
  - **Boolean**: A binary of 'True' or 'False'. This will be returned by the 'IsHomozygous' or 'SNPIsHomozygous' functions, 'IsSameSSR' or 'SNPMatch' functions, 'IsIncluded' or 'SNPIsIncluded' functions, 'AlleleShared' or 'SNPAlleleShared' functions, and 'IsChild', 'SNPIsChild', or 'MIsChild' functions. The result can be used as a conditional expression in an 'If' sentence.
  - **NUC**: a code of the deoxy-ribonucleotide or ribonucleotide.
  - **AA**: a single letter code of amino acid.
  - **SEQ**: nucleotide sequence of DNA or RNA, or amino acid sequence.
  - **Any**: any types of a continuous sequence of DNA, RNA, amino acid or else.

6. Precautions for data import

Automatic data format conversion is an outstanding capability of MS Excel to improving user experience. However, it often changes the already defined format unintendedly when importing data and may cause a trouble in the following analysis. Escape Excel (Welsh *et al.*, 2017) is an excellent tool to avoid unintended transformation of data by escaping text data before importing it into MS Excel. When the imported data returns an unintended result, confirm whether the data format was not altered by the automatic functionality of MS Excel. When you found the data altered, use Escape Excel before importing them.

7. Collaboration with R

Various R packages have been developed for the advanced analysis of genotype data. Although GUGS is not an R application, it is accessible to the data in R by integrating with RExcel system. Refer the RExcel website (http://rcom.univie.ac.at/) about how to install and use it in your system.

8. License

Table 1. Summary of the available data types in GUGS

Data type	Input	Output	Example
SSR genotype	O	O	100/110, 200/200 ...
SSR allele	O	O	100, 110, 200 ...
SNP genotype	O	O	A/G, CG ...
SNP allele	O	O	A, G, C ...
HMP	O	-	A single-letter genotype code for HapMap project
M	O	O	A single-letter genotype for MapMaker/JoinMap (A/B/H)
CODE	O	-	A single-letter genotype for the simplified SNP used in Illumina Genome Studio or else (A/B/H)
Seg	O	O	Segregation mode; BC1, F2, F1
CP type	O	O	A segregation type for CP mode of JoinMap; <abxcd>, <efxeg>, <hkxhk>, <lmxll>, <nnxnp>
Numeric	O	O	Allowed difference for SSR matching, or returned value of various functions
Boolean	-	O	‘True’ or ‘False’
NUC	O	O	A code of the deoxy-ribonucleotide (A, C, G, T) or ribonucleotide (A, C, G, U)
AA	O	O	A single letter code of amino acid
SEQ	O	O	Nucleotide sequence of DNA or RNA, or amino acid sequence
Any	O	O	Any types of sequence (DNA, RNA, amino acid or else)

O: Available for input data, output data, or both.

-: Not available for input data or output data.

9. Implemented functions

1) Data format conversion

Functions in this category allow to normalize genotype data, convert the absolute size to the relative size of an SSR genotype, insert a separator into an SNP genotype, convert a single-letter genotype into the pseudo-SNP format and extract the difference between two alleles of SSR genotype. It also allows converting Illumina's single-letter genotype format into an SNP genotype, and a converting single-letter HapMap code into a real SNP genotype.

a. Normalizing genotype data

These functions evaluate a designated string and then return a formatted SSR/SNP/M genotype.

SSR: The smaller allele will be sorted to the left side for an SSR genotype. A single allele will be converted to a homozygous genotype. Extra spaces will be removed.

SNP: The separator (a slash in default) will be removed, and then the alleles will be sorted according to alphabetical order. All small letters will be capitalized.

M: A small character will be capitalized; the initial character will be selected from multiple characters, and the remaining letters will be discarded. A numeric value or 'Null' is not allowed, in which case, an 'X' will be returned.

Data type	Usage	Return value
SSR	=NormSSR(Raw **SSR**)	Normalized SSR
SNP	=NormSNP(Raw **SNP**)	Normalized SNP
M	=NormMC(Raw **M**)	Normalized M

Tips: Assessing an ambiguous data before the evaluation is recommended for critical analysis.

b. Data conversion

These functions convert SSR, SNP, or M genotypes.

Data type	Usage	Return value
SSR	=SSRtoRelSize(**Case SSR**, **Ref SSR**) Converts alleles of an SSR genotype (**Case SSR**) to the relative values against the reference genotype (**Ref SSR**).	SSR genotype of alleles for relative size to the reference genotype
SNP	=SNPwithSEP(**SNP**, **S**) Converts SNP genotype (**SNP**) without a separator to the genotype with a separator (**S**).	SNP genotype delimited individual allele with the separator
M	=M2SNP(**M**) Converts a single-letter genotype (**M**) to an SNP-like two-letter genotype (AA, AB, or BB).	Formatted to SNP-like genotype

Tips: The ‘SSRtoRelSize’ function is useful for comparing SSR genotype data that were obtained from the different analysis. Any genotype data analysis software requires to separate SNP allele by a slash, and ‘SNPwithSEP’ function supports format conversion for these SNP data.

c. Obtain allele size difference of an SSR marker

This function returns a value of the size difference of two alleles of an SSR genotype (**SSR**).

Data type	Usage	Return value
SSR	=SSRDiff(SSR)	Numerical value of relative size between alleles in a designated SSR genotype

d. Convert Illumina's single-letter genotype to an SNP genotype

This function converts Illumina's single-letter genotype (**CODE**) for a two-letter SNP genotype by referring to the reference SNP genotype (**REF**) and strand information (**STRAND**).

Data type	Usage	Return value
SNP	=InterpretSNP(REF, STRAND, CODE)	SNP

REF:reference SNP genotype, **STRAND**:”TOP” or “BOT”, **CODE**:Illumina’s single-letter genotype
Tips: The obtained SNP genotype will vary according to the **STRAND** value (see the definition manual by Illumina).

Ref. http://www.illumina.com/documents/products/technotes/technote_topbot.pdf

e. Convert HapMap's single-letter genotype to an SNP genotype

This function converts the single-letter genotype of HapMap style (**HMP**) to a two-letter SNP genotype according to IUPAC format.

Data type	Usage	Return value
M	=HMP2SNP(HMP)	SNP (IUPAC format)

2) Basic analysis

Functions in this category allow to extract allele of an SSR or SNP genotype, examine homozygosity, determine the identity between two genotypes of SSR or SNP format, examine allele inclusion in the given genotype, evaluate the ploidy of an SSR genotype, separate all alleles of SSR or SNP genotypes and extract the shared allele between two SSR or SNP genotypes.

a. Right-side allele of genotype

These functions return the right-side allele of a designated SSR or SNP genotype. These functions do not sort the alleles before evaluation (See 'Normalize genotype data' section).

Obtain right-side allele of genotype

Data type	Usage	Return value
SSR	=RightAllele(SSR)	Right-side SSR allele
SNP	=SNPRightAllele(SNP)	Right-side SNP allele
M	N/A	N/A

Tips: Use the 'RightAllele' function in combination with the 'NormSSR' function will return the larger allele of an SSR genotype:

= RightAllele(NormSSR(**SSR**))

Similar technique will return the larger allele of an SNP genotype in alphabetical order:

= SNPRightAllele(NormSNP(**SNP**))

b. Left-side allele of genotype

These functions return the left-side allele of a designated SSR or SNP genotype. These functions do not sort the alleles before evaluation (See 'Normalize genotype data' section).

Obtain left-side allele of genotype

Data type	Usage	Return value
SSR	=LeftAllele(SSR)	Left-side SSR allele
SNP	=SNPLeftAllele(SNP)	Left-side SNP allele
M	N/A	N/A

Tips: Use the 'LeftAllele' function in combination with the 'NormSSR' function will return the smaller allele of an SSR genotype:

= RightAllele(NormSSR(**SSR**))

Similar technique will return the smaller allele of an SNP genotype in alphabetical order:

= SNPLeftAllele(NormSNP(**SNP**))

c. Examine homozygosity

These functions examine the homozygosity of a designated genotype (**SSR** or **SNP**) then return the result as a Boolean value ('True' will be returned when a designated genotype is homozygous; 'False' when heterozygous).

Examine homozygosity		
Data type	Usage	Return value
SSR	= IsHOMOZygous(SSR)	Boolean
SNP	= SNPIsHOMOZygous(SNP)	Boolean
M	N/A	N/A

Tips: The homozygosity of an M genotype can be achieved by searching for 'H' with the built-in function of MS Excel. Using this function with the **NOT** operand will examine the heterozygosity of the genotype.

=**NOT**(IsHomozygous("100/120")) will return 'TRUE'.

d. Examine genotype identity

These functions examine whether the two designated SSR or SNP genotypes are identical and then return the result as a Boolean value. The given genotype data will be normalized before the identity check. In the case of SSR genotype data, allele sizes up to the 'Diff' disagreement are allowed; however, these designated SSR genotypes should be kept for their heterozygosity or homozygosity even if the assigned '**Diff**' value allowed coinciding their apparent genotypes to each other.

Examine genotype identity		
Data type	Usage	Return value
SSR	=IsSameSSR(**SSR1**, **SSR2**, **Diff**) **SSR1** and **SSR2** are SSR genotypes to be compared, **Diff** for allowed maximum difference between two alleles.	Boolean
SNP	=SNPMatch(**SNP1**, **SNP2**) **SNP1** and **SNP2** are SNP genotypes to be compared.	Boolean
M	N/A	N/A

Tips: Consider setting an appropriate **Diff** value (allowed maximum difference between two alleles) for the SSR genotype to obtain the anticipated result.

= IsSameSSR("100/110", "101/111", 2) will return '**TRUE**'

= IsSameSSR("100/110", "101/111", 0) will return '**FALSE**'

e. Test an allele is included in genotype data

These functions examine whether a designated allele is included in the SSR or SNP genotype, then return the result as a Boolean value. The argument should be allele, not genotype.

Examine an allele is included		
Data type	Usage	Return value
SSR	= IsIncluded(SSR, allele)	Boolean
SNP	= SNPIsIncluded(SNP, allele)	Boolean
M	N/A	N/A

Tips: Multiple letters can be used for the "allele" to the SNPIsIncluded function, but only the initial letter will be valid for the evaluation. The test for an M genotype can be achieved by the built-in search function of MS Excel.

f. Examine ploidy from an SSR genotype

This function estimates the ploidy of an SSR genotype from the number of separators in the genotype. This function is available only for an SSR genotype. The slash is used as the separator.

Obtain ploidy of an SSR genotype		
Data type	Usage	Return value
SSR	= SSRPloidy(SSR)	An integer that represents the ploidy
SNP	N/A	N/A
M	N/A	N/A

Tips: This function counts the number of separators in the genotype data.

g. Split genotype into alleles

These functions split an SSR or SNP genotype into alleles and then return them in an array of cells separately. The split allele will be returned as array formulas; therefore, the user should assign cells to store each allele when using these functions (specify a region to retrieve the returned array, enter the formula, then press Ctrl + Shift + Enter at once). These functions are available for **SSR** or **SNP** genotypes.

Split genotype into alleles		
Data type	Usage	Return value
SSR	= SSRAllele(SSR)	Array formula of SSR alleles
SNP	= SNPAllele(SNP)	Array formula of SNP alleles
M	N/A	N/A

Tips: Refer the following link about the array formula in MS Excel.

https://support.office.com/en-us/article/Guidelines-and-examples-of-array-formulas-7d94a64e-3ff3-4686-9372-ecfd5caa57c7

h. Find a shared allele between two genotypes

These functions find a shared allele between two genotypes (**SSR1** and **SSR2** for SSR genotype, **SNP1** and **SNP2** for SNP genotype) and then return the shared allele found. If no allele is shared between the designated genotypes, ‘0’ (SSR) or ‘Null’ (SNP) will be returned. When both alleles are shared between them, the left-side allele will be returned.

Find a shared allele between two GTs

Data type	Usage	Return value
SSR	= SSRSharedAllele(SSR1, SSR2)	The shared SSR allele
SNP	= SNPSharedAllele(SNP1, SNP2)	The shared SNP allele
M	N/A	N/A

Tips: These functions are useful to examine a possible parentage or sibling relationship between two individuals.

3) Linkage analysis

Functions in this category involve to estimate a segregation mode of offspring from the genotype data of parents and convert the genotype of offspring for evaluation with linkage analysis software such as MapMaker (Lander *et al.*, 1987), JoinMap (Stam, 1993), or their equivalent software.

a. Estimate a segregation mode of two individuals for CP mode

These functions return a text to represent a segregation mode according to the CP mode of JoinMap estimated from the genotypes of two individuals (**SSR1** and **SSR2** for SSR genotype, **SNP1** and **SNP2** for SNP genotype). The return value will be "<abxcd>", "<efxeg>", "<hkxhk>", "<lmxll>" or "<nnxnp>". The values "F1" or "NP" will be returned when two genotypes are homozygous for difference allele ("F1") or identical for their genotypes, and no segregation is anticipated ("NP").

Estimate segregation mode of two individuals for CP mode

Data type	Usage	Return value
SSR	= SSR2CPType(SSR1, SSR2)	Text for segregation type
SNP	= SNP2CPType(SNP1, SNP2)	Text for segregation type

Tips: Use the 'SSR2CPGT' or 'SNP2CPGT' functions to obtain the converted genotype for the CP mode of JoinMap.

b. Convert a genotype for CP mode

These functions convert a raw genotype of offspring to an abstracted genotype according to the segregation type for the CP mode of JoinMap and genotypes of the parents. The return value will be obtained according to the segregation mode and the genotypes of parents. The values "-/-" will be returned when two genotypes are homozygous for difference allele (F1) or identical for their genotypes and no segregation is anticipated. The value "?/?" will be returned when genotypes of the parents and offspring do not agree with others.

Convert an offspring genotype for CP mode

Data type	Usage	Return value
SSR	= SSR2CPGT(SSR1, SSR2, Type, offspring SSR)	Abstracted genotype of offspring
SNP	= SNP2CPGT(SNP1, SNP2, Type, offspring SNP)	Abstracted genotype of offspring

SSR1/SSR2 or **SNP1/SNP2**: SSR or SNP genotypes of assigned parents, **Type**: segregation type for the CP mode of JoinMap anticipated from the assigned parents, **offspring SSR** or **offspring SNP**: SSR or SNP genotype of an offspring to be converted.

Tips: Use '**SSR2CPType**' or '**SNP2CPType**' functions to estimate a segregation mode for the CP mode of JoinMap.

c. Estimate a segregation mode of parents for BC1 or F2 mode

These functions estimate a segregation mode of offspring from the genotypes of two parental genotypes for the BC1 or F2 mode of JoinMap. The return value will be "F2" for F2 segregation, "BC1A" or "BC1B" for backed cross segregation (BC1) of the parent A (SSR1) or the parent B (SSR2), "F2D" for dominant type F2 segregation. The values "-" will be returned when two genotypes are homozygous for difference allele (F1) or identical for their genotypes and no segregation is anticipated.

Estimate segregation mode for BC1 or F2 mode

Data type	Usage	Return value
SSR	= SSR2SegType(SSR1, SSR2)	Text for segregation mode
SNP	= SNP2SegType(SNP1, SNP2)	Text for segregation mode

SSR1/SSR2 or **SNP1/SNP2**: SSR or SNP genotypes of assigned parents

Tips: Use the 'SSR2BC1' or 'SNP2BC1GT' functions to obtain the converted genotype for BC1 segregation, or use the 'SSR2F2GT' or 'SNP2F2GT' functions to obtain the converted genotype for F2 segregation.

d. Convert a genotype for BC1 mode

These functions convert a raw genotype of offspring for the BC1 mode of JoinMap. The return value will be "a", "b" or "h" for homozygote of parent A or B, or a heterozygote of parent A and B. The value "-" will be returned when two genotypes are not valid for BC1 segregation. The value "X" will be returned when invalid allele was assigned to either one of the assigned genotypes.

Convert an offspring genotype for BC1 mode

Data type	Usage	Return value
SSR	SSR2BC1GT(SSR1, SSR2, offspring SSR)	Text for offspring genotype
SNP	SNP2BC1GT(SNP1, SNP2, offspring SNP)	Text for offspring genotype

SSR1/SSR2 or **SNP1/SNP2**: SSR or SNP genotypes of assigned parents; **offspring SSR** or **offspring SNP**: SSR or SNP genotypes of an offspring to be converted.

Tips: Use the 'SSR2SegType' or 'SNP2SegType' functions to estimate the segregation type for BC1 or F2.

e. Convert a genotype for F2 mode

These functions convert a raw genotype of offspring for the F2 mode of JoinMap. The return value will be "a", "b" or "h" for homozygote of parent A or B, or a heterozygote of parent A and B. The value "-" will be returned when two genotypes are not valid for BC1 segregation. The value "X" will be returned when invalid allele was assigned to either one of the assigned genotypes.

Convert genotype for F2 mode

Data type	Usage	Return value
SSR	SSR2F2GT(SSR1, SSR2, offspring SSR)	Text for offspring genotype
SNP	SNP2F2GT(SNP1, SNP2, offspring SNP)	Text for offspring genotype

SSR1/SSR2 or **SNP1/SNP2**: SSR or SNP genotypes of assigned parents; **offspring SSR** or **offspring SNP**: SSR or SNP genotypes of an offspring to be converted.

Tips: Use the 'SSR2SegType' or 'SNP2SegType' functions to estimate the segregation type for BC1 or F2.

f. Estimate a segregation of an M-formatted genotype

This function will return "TRUE" when two M-formatted parental genotypes (**M1** and **M2**) will segregate for their offspring.

Estimate segregation of an M-formatted genotype		
Data type	Usage	Return value
M	= MWillSegregate(M1, M2)	Boolean

Tips: Use this function to identify the valid data for linkage analysis quickly. Use 'MSegregateType' to estimate the segregation type for BC1 or F2.

g. Estimate a segregation mode of two parental genotypes according to BC1 or F2

This function estimates a mode of segregation that is anticipated with the assigned M-formatted parental genotypes (**M1** and **M2**) according to a BC1- or F2-style genotype. The return value will be "F2" for F2 segregation, "BC1A" or "BC1B" for backed cross segregation (BC1) of the parent A or the parent B. The values "-" will be returned when no valid segregation is anticipated or an assigned genotype is inconsistent to others.

Estimate segregation mode of two parental genotypes according to BC1 or F2		
Data type	Usage	Return value
M	= MSegregateType(M1, M2)	Text for segregation mode

4) Dataset analysis

Functions in this category allow to estimate unique alleles or genotypes in the dataset, extract an allele or genotype at a designated rank in the dataset, and count the ratio of the matched allele or genotype in the dataset.

a. Number of unique alleles in a designated dataset

These functions return the number of unique alleles in a designated region. The argument is not a genotype but a region (eg., "A1:A100") that consists of a set of genotype data. Four types of genotype (SSR, SNP, M or allele) are available for the evaluation. The return value is the number of the unique alleles included in the assigned region.

Number of unique alleles

Data type	Usage	Return value
SSR	= UniqSSRAlleles(A region for SSRs)	An integer number of alleles
SNP	= UniqSNPAlleles(A region for SNPs)	An integer number of alleles
M	= UniqMAlleles(A region for Ms)	An integer number of alleles
Allele	= UniqAlleles(A region for alleles)	An integer number of alleles

Tips: Functions 'SSRAllele' or 'SNPAllele' will split all genotypes to an array of the allele.

b. Number of unique genotypes in a designated dataset

These functions return the number of unique genotypes in a designated region. The argument is not a genotype but a region (eg., "A1:A100") that consists of a set of genotype data. Three types of genotype (SSR, SNP or M) are available for the evaluation. The return value is the number of the unique genotypes included in the assigned region.

Number of unique genotypes

Data type	Usage	Return value
SSR	= UniqSSRGTs(A region for SSRs)	An integer number of genotypes
SNP	= UniqSNPGTs(A region for SNPs)	An integer number of genotypes
M	= UniqMGTs(A region for Ms)	An integer number of genotypes
Allele	N/A	N/A

Tips: Use 'UniqAlleles' function for the M genotype.

c. A unique allele in a designated dataset

These functions return the unique allele that appeared at index position in the dataset. The argument is not a genotype but a region (eg., "A1:A100") that consists of a set of genotype data. Four types of genotype (SSR, SNP, M or allele) are available for the evaluation. The **index** represents the order of a unique allele appeared in the dataset. 'Null' will be returned when the assigned index exceeds the maximum number of alleles.

The ith unique allele in a designated dataset		
Data type	Usage	Return value
SSR	= GetUniqSSRAllele(A region for SSRs, index)	A unique allele appeared in the region defined by the index
SNP	= GetUniqSNPAllele(A region for SNPs, index)	A unique allele appeared in the region defined by the index
M	= GetUniqMAllele(A region for Ms, index)	A unique allele appeared in the region defined by the index
Allele	= GetUniqAllele(A region for alleles, index)	A unique allele appeared in the region defined by the index

Tips: The maximum number of alleles in a particular region can be obtained by the functions 'UniqSSRAlleles', 'UniqSNPAlleles', 'UniqMAlleles', or 'UniqAlleles'.

d. A unique genotype in a designated dataset

These functions return the unique genotype that appeared at index position in the dataset. The argument is not a genotype but a region (eg., "A1:A100") that consists of a set of genotype data. Three types of genotype (SSR, SNP or M) are available for the evaluation. The **index** represents the order of a unique genotype appeared in the dataset. 'Null' will be returned when the assigned index exceeds the maximum number of genotypes.

A unique genotype in a designated dataset		
Data type	Usage	Return value
SSR	= GetUniqSSRGT(A region for SSRs, index)	A unique genotype appeared in the region defined by the index
SNP	= GetUniqSNPGT(A region for SNPs, index)	A unique genotype appeared in the region defined by the index
M	= GetUniqMGT(A region for Ms, index)	A unique genotype appeared in the region defined by the index
Allele	N/A	N/A

Tips: Use 'GetRepAlleles' function for the M genotype.

e. Ratio of matched genotypes by pairwise comparison between the two datasets

These functions compare the genotypes in two designated regions (**region1** and **region2**) pairwisely and then return the ratio of the matched genotypes in the region. The arguments are not genotypes but regions (eg., "A1:A100"). Four types of genotype (SSR, SNP, M or allele) are allowed for the evaluation. The return value will be ranged from 0 to 1. 'Zero' will be returned when the sizes of the designated regions are not identical.

The ratio of matched genotypes by pairwise comparison between the two datasets		
Data type	Usage	Return value
SSR	= SSRMatchedRatio(Region1, region2 for SSRs)	Ratio of the pairwisely matched genotypes
SNP	= SNPMatchedRatio(Region1, region2 for SNPs)	Ratio of the pairwisely matched genotypes
M	= MMatchedRatio(Region1, region2 for Ms)	Ratio of the pairwisely matched genotypes
Allele	= MatchedRatio(Region1, region2 for alleles)	Ratio of the pairwisely matched genotypes

Tips: These functions are available for quick evaluation of the similarity between two datasets. They do not compare each genotype in these regions for all-by-all but pairwisely. The 'MMatchedRatio' function is restricted to evaluate for three genotypes ('A', 'B', or 'H') and is not applicable for an allele of SSR or SNP genotype. The 'MatchedRatio' function does not have such restriction, and any types of an allele of SSR or SNP genotype is acceptable.

f. Ratio of matched alleles between the two datasets by pairwise comparison

These functions compare each allele in two designated regions (**region1** and **region2**) pairwisely and then return a value for the ratio of the matched alleles in the region. The arguments are not genotypes but regions (eg., "A1:A100") that consist of two sets of genotype data to be compared. Three types of genotype (SSR, SNP or M) are available for the evaluation. The return value will be ranged from 0 to 1. 'Zero' will be returned when the sizes of the designated regions are not identical.

The ratio of matched alleles by pairwise comparison between the two datasets

Data type	Usage	Return value
SSR	= SSRSharedRatio(Region1, region2 for SSRs)	Ratio of the shared alleles
SNP	= SNPSharedRatio(Region1, region2 for SNPs)	Ratio of the shared alleles
M	= MSharedRatio(Region1, region2 for Ms)	Ratio of the shared alleles
Allele	N/A	N/A

Tips: These functions are useful for quick evaluation of allele sharing between two dataset. They do not compare each allele in these regions for all-by-all but pairwisely. The 'MSharedFreq' function is restricted to evaluate for three genotypes ('A', 'B', or 'H') and is not applicable for an allele of SSR or SNP genotype. The 'MSharedFreq' function considers that 'H' is heterozygous of 'A' and 'B'. Use 'MatchedRatio' function to evaluate alleles for this purpose.

5) Frequency analysis

Functions in this category allow estimating allele frequencies or diversity of population data, observed or expected heterozygosity, polymorphic information content (PIC), match probability, the power of discrimination, genetic diversity, and the probability of showing an identical genotype for different individuals in a population dataset according to the theorem of Ukai.

a. Frequency of a particular allele in a designated dataset

These functions return the frequency of a particular allele from a set of genotype data in a designated region. The first argument is not a genotype but regions (eg., "A1:A100") that consist of sets of genotype data to be evaluated. The second argument is an allele of interest. Four types of genotype (SSR, SNP, M or allele) are available for the evaluation. The return value will be ranged from 0 to 1. 'Zero' will be returned when the assigned allele is not observed in the region.

The frequency of a particular allele in a designated dataset

Data type	Usage	Return value
SSR	= SSRAlleleFreq(A region for SSRs, allele)	Frequency of allele
SNP	= SNPAlleleFreq(A region for SNPs, allele)	Frequency of allele
M	= MAlleleFreq(A region for Ms, allele)	Frequency of allele
Allele	= AlleleFreq(A region for alleles, allele)	Frequency of allele

Tips: Use the 'UniqSSRAlleles', 'UniqSNPAlleles', 'UniqMAlleles', or 'UniqAlleles' functions to obtain the number of unique alleles. The individual alleles can be obtained by the 'GetRepSSRAllele', 'GetRepSNPAllele', 'GetRepMAllele', or 'GetRepAllele' functions.

b. Frequency of a particular genotype in a designated dataset

These functions return the frequency of a particular genotype from a set of genotype data in a designated region. The first argument is not a genotype but regions (eg., "A1:A100") that consist of sets of genotype data to be evaluated. The second argument is a genotype of interest. Three types of genotype (SSR, SNP or M) are available for the evaluation. The return value will be ranged from 0 to 1. 'Zero' will be returned when the assigned allele is not observed in the region.

Genotype frequency

Data type	Usage	Return value
SSR	= SSRGTFreq(A region for SSRs, genotype)	Frequency of genotype
SNP	= SNPGTFreq(A region for SNPs, genotype)	Frequency of genotype
M	= MGTFreq(A region for Ms, genotype)	Frequency of genotype
Allele	---	---

Tips: The 'AlleleFreq' function is available for genotype. The number of unique genotypes can be obtained by the 'UniqSSRGTs', 'UniqSNPGTs', or 'UniqMGTs' functions. The individual genotypes can be obtained by the 'GetRepSSRGT', 'GetRepSNPGT', or 'GetRepMGT' functions.

c. Observed heterozygosity (*Ho*)

These functions estimate the observed heterozygosity for a set of genotypes in a designated region. The *H*o value will range from 0 to 1. This *Ho* represents the ratio of the heterozygous genotypes in the designated region.

Observed heterozygosity		
Data type	Usage	Return value
SSR	= SSRHo(A region for SSRs)	Observed heterozygosity
SNP	= SNPHo(A region for SNPs)	Observed heterozygosity
M	= MHo(A region for Ms)	Observed heterozygosity

d. Expected heterozygosity (*H*e)

These functions estimate the expected heterozygosity for a set of genotypes in a designated region. The *H*e value will range from 0 to 1.

Observed heterozygosity		
Data type	Usage	Return value
SSR	= SSRHZ(A region for SSRs)	Expected heterozygosity
SNP	= SNPHZ(A region for SNPs)	Expected heterozygosity
M	= MHZ(A region for Ms)	Expected heterozygosity

Tips: These functions estimate the heterozygosity according to Equation (8.1) of Nei, as below (Nei, 1987). (Nei defined the expected heterozygosity as *h*.) This is not an unbiased estimator of the expected heterozygosity. The unbiased estimator of the expected heterozygosity can be obtained by the GD or GD2 functions.

$$He = h = 1 - \sum_{i=1}^{m} x^2$$

e. Polymorphic information content (PIC)

These functions estimate the polymorphic information content (PIC) of a DNA marker from a set of the marker genotypes in a designated region. PIC is the probability to infer whether the allele of an offspring derived from either one of the parents. The PIC value will range from 0 to 1.

Polymorphic information content		
Data type	Usage	Return value
SSR	= SSRPIC(A region for SSRs)	PIC value for the region
SNP	= SNPPIC(A region for SNPs)	PIC value for the region
M	= MPIC(A region for Ms)	PIC value for the region

Tips: These functions estimate the PIC according to the following formula:

$$PIC = 1 - \sum_{i=1}^{m} {p_i}^2 - \sum_{j>i}^{n} 2(p_i p_j)^2$$

f. Match probability (PM)

These functions estimate the match probability (PM) for a set of genotypes in a designated region. PM is the probability that the two randomly selected individuals will have identical genotypes. The PM value will range from 0 to 1.

Match probability		
Data type	Usage	Return value
SSR	= SSRPM(Region for SSRs)	PM value for the region
SNP	= SNPPM(Region for SNPs)	PM value for the region
M	= MPM(Region for Ms)	PM value for the region

Tips: These functions estimate the match probability according to Goodwin, Linacre, and Hadi (Goodwin *et al.*, 2011) using the following formula:

$$pM = \sum_{k=1}^{m} {p_k}^2$$

g. Power of discrimination (PD)

These functions estimate the power of discrimination (PD) for a set of genotypes in a designated region. The PD value will be ranged from 0 to 1. PD is the probability that the two randomly selected individuals will have a different genotype.

Power of discrimination (PD)		
Data type	Usage	Return value
SSR	= SSRPD(A region for SSRs)	PD value for the region
SNP	= SNPPD(A region for SNPs)	PD value for the region
M	= MPD(A region for Ms)	PD value for the region

Tips: These functions estimate the power of discrimination according to Goodwin, Linacre, and Hadi (Goodwin *et al.*, 2011) using the following formula:

$$pD = 1 - p\mathrm{M}$$

h. Unbiased estimator of expected heterozygosity for a random population (GD)

These functions return the expected heterozygosity for a random cross population (GD) with a set of genotypes in a designated region. The GD value will be ranged from 0 to 1.

Unbiased estimator of expected heterozygosity for a random population (GD)		
Data type	Usage	Return value
SSR	= SSRGD(A region for SSRs)	GD value for the region
SNP	= SNPGD(A region for SNPs)	GD value for the region
M	= MPGD(A region for Ms)	GD value for the region

Tips: These functions estimate the unbiased heterozygosity for a random population according to Equation (8.4) of Nei (Nei, 1987). Nei assigned $\hat{h}$ for the gene diversity (GD) (Nei, 1987).

$$GD = \hat{h} = 2\mathrm{n}(1 - \sum \hat{x}_i^2)/(2n - 1)$$

i. Unbiased estimator of expected heterozygosity for a selfed population (GD2)

These functions return the unbiased expected heterozygosity for a selfed population (GD2) with a set of genotypes in a designated region. The GD2 value will be ranged from 0 to 1.

Unbiased estimator of expected heterozygosity for a selfed population (GD2)

Data type	Usage	Return value
SSR	= SSRGD2(A region for SSRs)	GD value for the region
SNP	= SNPGD2(A region for SNPs)	GD value for the region
M	= MPGD2(A region for Ms)	GD value for the region

Tips: The unbiased estimator of expected heterozygosity for a selfed population was also defined as $\hat{h}$ by Equation (8.5) of Nei (Nei, 1987):

$$GD2 = \hat{h} = \mathrm{n}(1 - \sum \hat{x}_i^2)/(n-1)$$

j. Direct product of genotype frequency (*f*0) according to Ukai's theory of varietal identification

This function returns a direct product for the particular genotypes in a population (f0) that are estimated separately according to Ukai (Ukai, 2004). The obtained f_0 value is then provided to estimate the probability P1 according to the theory by Ukai that indicates a randomly selected individual will give the same genotype set.

$$f_0 = \prod_{i=1}^{n} f_i$$

Tips: This is a common function of all genotype data.

k. Probability that a randomly selected individual will show a genotype identical to a particular individual

This function returns the probability of a randomly selected individual being identical in genotype to a particular individual in a population. This probability is estimated from the direct power for the particular genotypes in a population (f0) obtained separately with the 'Ukaif0' function. The P1 value is obtained according to the theory by Ukai (Ukai, 2004).

$$P_1 = 1 - (1 - f_0)^N$$

Tips: This is a common function of all genotype data.

6) Genetic data analysis

Functions in this category are valid to estimate parentage of offspring within a population and determine the probability to obtain an individual from a cross of designated parents in a population.

a. Allele sharing test (single parent test)

These functions examine whether two genotypes share the same allele and then return the result as a Boolean value (single parent test). The value will be 'TRUE' when two genotypes share at least one allele or 'False' when nothing is shared between them.

Allele sharing test

Data type	Usage	Return value
SSR	= AlleleShared(SSR1, SSR2)	Boolean
SNP	= SNPAlleleShared(SNP1, SNP2)	Boolean
M	= MAlleleShared(M1, M2)	Boolean

SSR1/SSR2, **SNP1/SNP2** or **M1/M2** are SSR, SNP or M genotypes to be compared.

Tips: According to Mendel's law of inheritance, all offspring should inherit a set of diploid alleles, one allele from each parent. Hence, any siblings will share 0, 1, or 2 alleles among them. This function is available to identify the parents or offspring with significant numbers of DNA markers.

b. Trio test (family test)

These functions examine whether a particular genotype of an offspring is consistent with those of two parents and then return the result as a Boolean value. The value will be 'True' when the given genotype is consistent with being an offspring of the parents or 'False' when the genotypes are inconsistent with belonging to a family.

Trio test

Data type	Usage	Return value
SSR	= IsChild(SSR of offspring, SSR1, SSR2 of parents)	Boolean
SNP	= SNPIsChild(SNP of offspring, SNP1, SNP2 of parents)	Boolean
M	= MIsChild(M of offspring, M1, M2 of parents)	Boolean

Tips: These functions evaluate whether each allele of the offspring was inherited from the proposed parents, according to Mendel's law of inheritance. Using multi-allelic genotype data, such as an SSR genotype is advantageous for inferring the parentage of the offspring than SNP or M genotypes. For example, a cross AB x CD will give offspring with an AC genotype, but an opposite cross AB x AC is not consistent with a proposed offspring with a CD genotype. This enables us to infer the correct combination of the parents and offspring. On the contrary, a cross with a bi-allelic marker genotype AB x AA will give offspring with an AB genotype, but the opposite cross AB x AB is also consistent with a possible offspring with an AA genotype. This means that bi-allelic genotype data has less power to infer the exact combination of parentage in most cases.

c. Probability of obtaining a particular offspring from parents

These functions return the probability of obtaining offspring with particular genotypes from two parental individuals for which their genotypes are known. These functions estimate $T(g_0|g_m, g_a)$ by Marshall (1998) (Marshall *et al.*, 1998) from the proposed genotypes of the offspring and parents. The result will be '0', '0.25', '0.5', '0.75', or '1' according to their genotypes. An integer value of the maximum allowed difference (Diff) can be assigned for the evaluation of SSR genotype.

Probability of obtaining a particular offspring from parents		
Data type	Usage	Return value
SSR	= SSRChildProbability(SSR of offspring, SSR1, SSR2 of parents, Diff)	Probability
SNP	= SNPChildProbability(SNP of offspring, SNP1, SNP2 of parents)	Probability
M	= MChildProbability(M of offspring, M1, M2 of parents)	Probability

Tips: Set 0 to **Diff** for SSR genotype data when a strict match is anticipated.

d. Probability of obtaining a particular offspring from a random mating population

These functions return the probability of obtaining an offspring of a particular genotype from the random mating of a designated population. The first argument is not a genotype but a region (eg., “A1:A100”) that consists of sets of genotype data to be evaluated. The second argument is a genotype of interest. Three types of genotype (SSR, SNP or M) are available for the evaluation. The return value will be ranged from 0 to 1. These functions estimate the P(g_o) of Marshall (1998) (Marshall *et al.*, 1998).

Probability of obtaining a particular offspring from a random mating population		
Data type	Usage	Return value
SSR	= SSRGTProbability(A region for SSRs, SSR)	Probability
SNP	= SNPGTProbability(A region for SNPs, SNP)	Probability
M	= MGTProbability(A region for Ms, M)	Probability

Tips: The value of probability will be varied depending on region or genotype.

e. Probability of obtaining a particular offspring from an individual of the population and an alleged parent

These functions estimate the probability of obtaining offspring with a particular genotype from an alleged parent and a randomly selected undefined alleged single parent in a population in the designated region. The first argument is a genotype of interest, and the second is the genotype of alleged parent. The third argument is not a genotype but a region (eg., “A1:A100”) that consists of sets of genotype data to be evaluated. Two types of genotype (SSR or SNP) are available for the evaluation. The return value will be ranged from 0 to 1. These functions estimate the T($g_0|g_m$) of Marshall (1998) (Marshall *et al.*, 1998).

Probability of obtaining a particular offspring from an individual of the population and an alleged parent		
Data type	Usage	Return value
SSR	= SSRParentageProbability(SSR of offspring, SSR of alleged parent, a region for population)	Probability
SNP	= SNPParentageProbability(SNP of offspring, SNP of alleged parent, a region for population)	Probability
M	N/A	N/A

Tips: The likelihood ratio whether the alleged parents are the true parents of the given offspring can be estimated according to Equation (3) of Jones and Ardren (Jones and Ardren, 2003) using the following formula as described by Shimizu et al. (Shimizu *et al.*, 2016):

$$L(H_1, H_2|g_S, g_P, g_O) = \frac{T(g_O|g_S, g_P)}{P(g_B)}$$

For SSR markers, the likelihood ratio is given by using these functions:

```
= SSRChildProbability(arguments) / SSRGTProbability(arguments)
```

The likelihood ratio for the alleged single parent of the offspring is estimated by Equation (2) of Jones and Ardren (Jones and Ardren, 2003) using the following formula as described by Shimizu et al. (Shimizu *et al.*, 2016):

$$L(H_1, H_2|g_S, g_O) = \frac{T(g_O|g_S)}{P(g_B)}$$

For SSR markers, the likelihood ratio is given by using these functions:

```
= SSRParentageProbability(arguments) / SSRGTProbability(arguments)
```

LOD score is the natural logarithm value of the likelihood ratio value.

7) Sequence manipulation

Functions in this category allow for converting, counting, formatting and searching for nucleotide or amino acid sequence motif to support DNA marker design.

a. Complementary nucleotide

This function returns complementary nucleotide (not for sequence).

Complementary nucleotide	
Usage	Return value
= compnuc(NUC)	A code of complementary deoxy-ribo nucleotide

NUC: a code of nucleotide (not sequence)

b. Complementary sequence

This function returns complementary sequence (not for nucleotide).

Complementary sequence	
Usage	Return value
= comp(SEQ)	Complementary sequence

SEQ: nucleic acid sequence

c. Reverse sequence

This function returns sequence in reverse direction.

Reverse sequence	
Usage	Return value
= reverse(SEQ)	Reversed sequence

SEQ: nucleic acid sequence

d. Reverse complementary sequence

This function returns complement sequence (not for nucleotide).

Reverse complement sequence	
Usage	Return value
= revcomp(SEQ)	Reversed complementary sequence

SEQ: nucleic acid sequence

e. Split sequence by inserting spaces at a particular interval

This function inserts a space to the sequence at every N characters interval. This function is available for DNA/RNA/amino acid sequence.

Split the sequence with space	
Usage	Return value
= splitseq(SEQ, interval)	Split sequence

SEQ: nucleic acid sequence, **interval**: length to insert a space.

f. Bracket a particular letter in the sequence

This function encloses a particular letter at the assigned position in the sequence with bracket. This function is available for DNA/RNA/amino acid sequence.

Split the sequence with space	
Usage	Return value
= bracket(SEQ, position)	Bracketed sequence

SEQ: nucleic acid sequence, **position**: a particular position to be enclosed in brackets.

g. Remove extra code in the sequence

This function removes all extra code (space, bracket, slash or parenthesis) in the sequence. This function is available for DNA/RNA/amino acid sequence.

Split the sequence with space	
Usage	Return value
= shrink(SEQ)	Shrinked sequence

SEQ: nucleic acid sequence.

h. Convert a DNA code to RNA code

This function converts a code of deoxy-ribonucleotide to a code of ribonucleotide. This function is valid for DNA sequence.

Convert a DNA code to RNA code	
Usage	Return value
= DNA2Rnuc(NUC)	A code of ribo nucleotide

NUC: a code of deoxy-ribonucleotide (not sequence)

i. Convert an RNA code to DNA code

This function converts a code of deoxy-ribonucleotide to a code of ribonucleotide. This function is valid for DNA sequence.

Convert an RNA code to DNA code	
Usage	Return value
= RNA2Dnuc(NUC)	A code of deoxy-ribo nucleotide

NUC: a code of ribo nucleotide (not sequence)

j. Convert DNA sequence to RNA sequence

This function converts T/t to U/u in the sequence. This function is valid for DNA sequence.

Convert a DNA sequence to RNA sequence	
Usage	Return value
= toRNA(SEQ)	Ribo nucleic acid sequence

SEQ: deoxy-ribo nucleic acid sequence (not a nucleotide).

k. Convert RNA sequence to DNA sequence

This function converts T/t to U/u in the sequence. This function is valid for DNA sequence.

Convert an RNA sequence to DNA sequence	
Usage	Return value
= toDNA(SEQ)	Deoxy-ribo nucleic acid sequence

SEQ: ribo nucleic acid sequence (not a nucleotide).

l. Format the sequence by inserting line feed at a particular interval

This function inserts a line feed at every Nth position in the sequence. This function is available for DNA/RNA/amino acid sequence.

Format the sequence by insert line feed at same interval	
Usage	Return value
= fold(SEQ, interval)	Formatted sequence

SEQ: DNA/RNA/amino acid sequence. **Interval**: duration to insert a line feed.

m. Extract 5' nucleotide sequence

This function extracts 5' nucleotide sequence of the assigned length. The 5' end nucleotide will be returned when the length was not defined. This function is valid for the nucleic acid sequence (DNA/RNA).

Extract 5' nucleotide sequence	
Usage	Return value
= clip5(SEQ, length)	Extracted sequence

SEQ: DNA/RNA sequence. **Length**: distance from the 5’ end.

n. Extract 3' nucleotide sequence

This function extracts 5' nucleotide sequence of the assigned length. The 3' end nucleotide will be returned when the length was not defined. This function is valid for the nucleic acid sequence (DNA/RNA).

Extract 3' nucleotide sequence	
Usage	Return value
= clip3(SEQ, length)	Extracted sequence

SEQ: DNA/RNA sequence. **Length**: distance from the 3' end.

o. Convert nucleotide sequence in particular reading frame to amino acid sequence

This function converts a nucleotide sequence of the assigned reading frame to amino acid sequence. This function is valid for the nucleic acid sequence (DNA/RNA).

Convert nucleotide sequence to amino acid sequence	
Usage	Return value
= nuc2aa(SEQ (,frame))	Amino acid sequence

SEQ: DNA/RNA sequence. **Frame**: reading frame (integer: optional; set 1 when not assigned).

p. Nucleotide composition and sequence length

This function reports the composition of nucleotides and the length of the sequence. This function is valid for the nucleic acid sequence (DNA/RNA).

Convert a DNA sequence to RNA sequence	
Usage	Return value
= composition(SEQ)	A text of the nucleotide composition and the sequence length

SEQ: nucleic acid sequence (DNA/RNA).

q. Report the GC ratio of the nucleic acid sequence

This function returns the GC ratio of the nucleic acid sequence. This function is valid for the nucleic acid sequence (DNA/RNA).

GC ratio of the nucleic acid sequence	
Usage	Return value
= GCratio(SEQ)	GC ratio of the sequence

SEQ: nucleic acid sequence (DNA/RNA).

r. Count the occurrence of a motif sequence appeared in the target sequence

This function counts an occurrence of a motif sequence appeared in the target sequence after the defined position. This function is available for any types of sequence.

Count the occurrence of a motif in the target sequence	
Usage	Return value
= motifcount(SEQ, motif (,position))	Number of the motif sequence in the target sequence

SEQ: nucleic acid sequence (DNA/RNA). **Motif**: a sequence of interest. **Position**: the position in the target sequence to start with the evaluation (optional; set 1 when not assigned).

s. Find the first position of a motif sequence appeared in the target sequence

This function returns the first position of a motif sequence found in the target sequence after the defined position. This function is available for any types of sequence.

Count the occurrence of a motif in the target sequence	
Usage	Return value
= firstmotif(SEQ, motif (,position))	The first position of the motif sequence appeared in the target sequence after the position.

SEQ: nucleic acid sequence (DNA/RNA). **Motif**: a sequence of interest. **Position**: the position in the target sequence to start with the evaluation (optional; set 1 when not assigned).

t. List all positions of the motif sequence appeared in the target sequence

This function returns the list of all positions of the motif sequence found in the target sequence as a text. This function is available for any types of sequence.

List of all positions of the motif sequence appeared in the target sequence	
Usage	Return value
= findmotif(SEQ, motif)	List of positions of the motif sequence appeared in the target sequence

SEQ: nucleic acid sequence (DNA/RNA). **Motif**: a sequence of interest.

u. Mark the all motif sequence appeared in the target sequence

This function encloses the all motif sequence appeared in the target sequence with brackets. This function is available for any types of sequence.

Mark the all motif sequence appeared in the target sequence	
Usage	Return value
= markmotif(SEQ, motif)	The target sequence enclosed all motif sequence appeared with brackets.

SEQ: a sequence to be evaluated. **Motif**: a sequence of interest.

v. Match the query sequence to the reference sequence

This function compares the query sequence to the reference sequence from 5'-end then shows the matched character. This function is available for any types of sequence.

Mark the all motif sequence appeared in the target sequence	
Usage	Return value
= matchseq(Ref, QUERY)	1st line: REF sequence, 2nd line: matched QUERY sequence but "." for unmatched.

REF: reference sequence to be compared. **QUERY**: a sequence of interest.

Tips: Enable "*Wrap text*" functionality of the cell to get the expected result.

w. Ratio of the matched character in the query sequence to the reference sequence

This function compares the query sequence to the reference sequence from 5'-end then returns the ratio of the matched character between them. This function is available for any types of sequence.

Ratio of the matched character between the query sequence to the reference sequence	
Usage	Return value
= matchscore(Ref, Query)	The ratio of the matched character between the reference sequence to the query sequence.

REF: reference sequence to be compared. **Query**: a sequence of interest.

x. Prune the unnecessary sequence and select the sequence in a bracket

This function removes the extra sequence outside of bracket, then return the sequence within a bracket. The user can identify the one to be obtained when multiple brackets present in the sequence. This function is available for any types of sequence.

Prune the unnecessary sequence and select the sequence in a bracket	
Usage	Return value
= prune(SEQ (, order))	A sequence in the bracket within the SEQ at a position defined by ORDER.

REF: sequence to be evaluated. **Order**: the position of a bracket to be extracted (optional).

10. Availability

GUGS is available at https://github.com/tokurou/GUGS under the GPL3 license.

11. Citation

Shimizu, T. (2021) General Utilities for Genotyping Study (GUGS): A Comprehensive Application in Genotype and Sequence Data Manipulation. *JARQ* **55** (4), 333-339.

References

Goodwin,W. *et al.* (2011) An introduction to forensic genetics Wiley, West Sussex, UK.

Jones,A.G. and Ardren,W.R. (2003) Methods of parentage analysis in natural populations. *Mol. Ecol.*, **12**, 2511–2523.

Lander,E.S. *et al.* (1987) MAPMAKER: an interactive computer package for constructing primary genetic linkage maps of experimental and natural populations. *Genomics*, **1**, 174–181.

Marshall,T.C. *et al.* (1998) Statistical confidence for likelihood-based paternity inference in natural populations. *Mol. Ecol.*, **7**, 639–655.

Nei,M. (1987) Molecular Evolutionary Genetics Columbia University Press, New Nork, U.S.A.

Shimizu,T. *et al.* (2016) Hybrid origins of citrus varieties inferred from DNA marker analysis of nuclear and organelle genomes. *PLoS One*, **11**, e0166969.

Stam,P. (1993) Construction of integrated genetic linkage maps by means of a new computer package: Join Map. *Plant J.*, **3**, 739–744.

Ukai,Y. (2004) A theory for varietal identification of plant cultivars. *Nougyou Oyobi Engei*, **79**, 194–198.

Welsh,E.A. *et al.* (2017) Escape Excel: a tool for preventing gene symbol and accession conversion errors. *PLoS One*, **12**, e0185207.