

# msq module

The main functions of `msq` module.

## msq.example\_data()

Imports example data (genetic map, allele substitution effects for three milk traits, phased genotypes, and phenotypic information).

## msq.Datacheck(gmap, meff, gmat, group, indwt, progress=False)

Checks the data for errors, converts haplotypes to genotypes and stores relevant info as an object.

### Parameters

- `gmap` : pandas DataFrame providing genetic map(s)  
Index: RangeIndex  
Columns:  
Name: CHR, dtype: int64; chromosome number  
Name: SNPName, dtype: object; marker name  
Name: Position, dtype: int64; marker position in bp  
Name: group, dtype: float64; marker distance (cM) or reco rates
- `meff` : pandas DataFrame providing allele substitution or marker effects  
Index: RangeIndex  
Columns:  
Name: trait names, dtype: float64; no. of columns = no of traits
- `gmat` : pandas DataFrame providing phased genotypes  
Index: RangeIndex  
Columns:  
Name: ID, dtype: int64 or str; identification of individuals  
Name: haplotypes, dtype: object; must be biallelic
- `group` : pandas DataFrame providing phenotypic information  
Index: RangeIndex  
Columns:  
Name: group, dtype: object; group code of individuals, e.g., M, F  
Name: ID, dtype: int64 or str; identification of individuals
- `indwt` : list of index weights for each trait
- `progress` : bool, optional; print progress of the function if True

### Returns

- `class object` : input data as a class object

## msq.popcovmat(info, mposunit, method)

Derives population-specific covariance matrices.

### Parameters

- `info` : A class object created using the function "msq.Datacheck"
- `mposunit` : string containing "cM" or "reco".
- `method` : An integer with a value of 1 for Bonk et al.'s approach or 2 for Santos et al.'s approach

### Returns

- `list` : A list containing group-specific pop covariance matrices for each chr.

## msq.msvarcov\_g(info, covmat, sub\_id, progress=False)

Derives Mendelian sampling co(variance) and aggregate genotype.

### Parameters

- `info` : A class object created using the function "msq.Datacheck"
- `covmat` : A list of pop cov matrices created using "msq.popcovmat" function
- `sub_id` : pandas dataframe of one column containing ID numbers of specific individuals to be evaluated  
Index: RangeIndex (minimum of 2 rows)  
Columns:  
Name: ID, dtype: int64 or str; identification of individuals
- `progress` : bool, optional; print progress of the function if True

### Returns

- `df` : pandas dataframe containing the Mendelian sampling (co)variance and aggregate genotype

Note : If sub\_id is None, Mendelian (co-)variance will be estimated for all individuals. Otherwise, Mendelian (co-)variance will be estimated for the individuals in sub\_id

## msq.msvarcov\_gcorr(msvmc)

Standardizes Mendelian sampling covariances.

### Parameters

- `msvmc` : pandas dataframe containing the Mendelian sampling (co)variance derived using function "msq.msvarcov\_g"

### Returns

- `df` : pandas dataframe containing the Mendelian sampling correlations

## msq.selstrat\_g(selstrat, info, sub\_id, msvmc, throrconst)

Calculates selection criteria (GEBV, PBTI, or bvindex using gametic approach.

### Parameters

- `selstrat` : str  
A str containing any of GEBV, PBTI or bvindex
- `info` : A class object created using the function "msq.Datacheck"
- `sub_id` : pandas dataframe of one column containing ID numbers of specific individuals to be evaluated  
Index: RangeIndex (minimum of 2 rows)  
Columns:  
Name: ID, dtype: int64 or str; identification of individuals
- `msvmc` : pandas dataframe created using the function "msq.msvarcov\_g"
- `throrconst` : float  
If selstrat is PBTI, a throrconst of value 0.05 sets threshold at top 5% of GEBV.  
If selstrat is bvindex, throrconst is a constant.

### Returns

- `df` : pandas dataframe with estimated selection criteria for each trait and aggregate  
Index: RangeIndex  
Columns:  
ID, Group, trait names and Overall

Note : If selstrat is GEBV, None may be used for throrconst and msvmc. If sub\_id is None and selstrat is GEBV, GEBVs will be estimated for all individuals. However, if selstrat is not GEBV, the chosen selection criterion will be estimated for all individuals in msvmc data frame.

## msq.simmat\_g(info, covmat, sub\_id, chrinterest, save=False, stdsim=False, progress=False)

Compute similarity matrices using gametic approach.

### Parameters

- `info` : A class object created using the function "msq.Datacheck"
- `covmat` : A list of pop cov matrices created using "msq.popcovmat" function
- `sub_id` : pandas dataframe of one column containing ID numbers of specific individuals to be evaluated  
Index: RangeIndex (minimum of 2 rows)  
Columns:  
Name: ID, dtype: int64 or str; identification of individuals
- `chrinterest` : str or list of int  
list of chromosome numbers of interest or str with "all" or "none"
- `save` : bool, optional; write trait-specific sim mats to file if true
- `stdsim` : bool, optional; print write std sim mats to file if true
- `progress` : bool, optional; print progress of the task if true

### Returns

- `list` : list containing similarity matrices for each group

## msq.selstrat\_z(selstrat, info, sub\_idz, msvmc, throrconst, selmale, selfm, maxmale)

Calculate selection criteria (GEBV, PBTI, or bvindex) for zygotes.

### Parameters

- `selstrat` : str  
A str containing any of GEBV, PBTI or bvindex
- `info` : A class object created using the function "msq.Datacheck"
- `sub_idz` : pandas dataframe containing ID numbers of specific mate pairs to be evaluated  
Index: RangeIndex (minimum of 2 rows)  
Columns:  
Name: Male, dtype: int64 or str; identification of males  
Name: Female, dtype: int64 or str; identification of females
- `msvmc` : pandas dataframe created using the function "msq.msvarcov\_g"
- `throrconst` : float  
If selstrat is PBTI, a throrconst of value 0.05 sets threshold at top 5% of GEBV of the population.  
If selstrat is bvindex, throrconst is a constant.
- `selmale` : list  
list of two items. 1st item is the str coding for males as in phenotypic information. The 2nd item is a float representing x% of males to be used
- `selfm` : list  
list of two items. 1st item is the str coding for females as in phenotypic information. The 2nd item is a float representing x% of females to be used
- `maxmale` : integer of maximum number of allocations for males

### Returns

- `df` : pandas dataframe  
Index: RangeIndex  
Columns:  
MaleID, FemaleID, MaleIndex, FemaleIndex, trait names and Overall

Note : If selstrat is GEBV, None may be used for throrconst and msvmc. If sub\_idz is None and selstrat is GEBV, GEBVs will be estimated for all matepairs. However, if selstrat is not GEBV, the chosen selection criterion will be estimated for all individuals in msvmc data frame.

## msq.simmat\_z(info, sub\_idz, covmat, chrinterest, save=False, stdsim=False, progress=False)

Compute similarity matrices using zygotic approach for specific matepairs.

### Parameters

- `info` : A class object created using the function "msq.Datacheck"
- `sub_idz` : pandas dataframe containing ID numbers of specific mate pairs to be evaluated  
Index: RangeIndex (minimum of 2 rows)  
Columns:  
Name: Male, dtype: int64 or str; identification of males  
Name: Female, dtype: int64 or str; identification of females
- `covmat` : A list of pop cov matrices created using "msq.popcovmat" function
- `chrinterest` : str or list of int  
list of chromosome numbers of interest or str with "all" or "none"
- `save` : bool, optional; write trait-specific sim mats to file if True
- `stdsim` : bool, optional; print write std sim mats to file if True
- `progress` : bool, optional; print progress of the task if True

### Returns

- `list` : containing similarity matrices for each group