In all cases, only the first four characters of the keyword are read. One can use more for clarity. Keywords are case-insensitive and there is no pre-established order of the keywords.

1. common_basis – *generation of the common MO basis*

Keyword	Description	Default
Project	Prefix of the files that will be generated	unknown
Threshold	Threshold for linear dependencies	1.0e-6
Fragments	Number of fragments followed by a line with the number of states per fragment	1 1
Frozen	Number of frozen orbitals on each fragment	0
NoAverage	Disable the construction of average frozen orbitals and use the orbitals of state 1 instead	.false.
AllEpsilons	Print all eigenvalues of the overlap matrix	.false.
Extra info	Prints the dimension of the MO basis for different thresholds	.false.
Debug	Insane printing	.false.

Input example:

```
Project
 benzene
Threshold
 1e-5
Fragments
 2
 5 5
Frozen
 6 6
NoAverage
AllEpsilons
Extra info
```

Output files

- COMMONORB	vector file in OpenMolcas format, to be used as input for the transformation of the integrals with MOTRA
- Project_xyz.vec	vector files expressing the fragment electronic states in the common MO basis, to be used by GronOR

Input files

- INPORB.x_y	vector files in OpenMolcas format of the fragment electronic states. x refers to the fragment, y to the fragment state
- ONEINTx	one-electron integrals file of the fragments
- RUNFILx	RunFile of the fragments
- RUNFILE	RunFile of the full system

2. rdcho – *Reconstruction of the integrals from the transformed Cholesky vectors*

`$> rdcho x` where 'x' is the number of ranks used in the MOTRA calculation.

When 'x' is larger than 1 (that is, a parallel execution of the integral transformation with cholesky decomposition), the `_CHMOT1` file has to be constructed from the different `_CHMOTx` files, each containing part of the info. Add the following lines of `EMIL` code at the end of the input used for `MOTRA`

```
>>> COPY _CHMOT1 $CurrDir/_CHMOT1
>>> eval NPROCS = $MOLCAS_NPROCS - 1
>>> foreach L in (1 .. $NPROCS )
>>> shell cat tmp_$L/_CHMOT1 >> $CurrDir/_CHMOT1
>>> enddo
```

3. rdtraint – *Conversion of TRAINT to Project_xxx.two*

Keyword	Description	Default
Project	Prefix of the files that will be generated	project
Write labels	Labels are also written on the 2-el. integrals files	.false.
Filesize	Size of the 2-el. integrals files in MB (approx.)	~2 GB
Small integrals	Threshold for considering an integral small Note that all integrals are written to disk, just for informative purposes.	1.0e-10
Print level	x < 10 ; standard output 10 =< x < 20 ; debug info x >= 20 ; all integrals are written in output	1

Input example:

```
Project
 benzene
Write labels
Filesize
 1500
Small integrals
 1.0e-8
Print level
 1
```

rdtraint and common_basis can read from the same input file.

Output files

- Project_xyz.two	2-el. integrals files to be read by GronOR
- Project.one	1-el. integrals files to be read by GronOR
- Project.sym	system info, coordinates and basis set.

Input files

- ONEINT one-electron integrals file of the full system
- RUNFILE RunFile of the full system
- TRAONE Transformed 1-el. integrals file, from MOTRA
- TRAINT Transformed 2-el. integrals file, from MOTRA or rdcho

4. addvect – *Merge vectors of fragment A and B, or split AB in A and B*

Line oriented input

Line 1: Title for the new vector file (in case of merging), not used when splitting

Line 2: Name of vector file of fragment A to be merged

Line 3: Name of vector file of fragment B to be merged

Line 4: Order of the vectors, one of the following: baab, abba, baba, abab, bbaa, aabb. The first appearance of 'a' or 'b' refers to the occupied orbitals, the second to virtuals.

Line 5: 'add' or 'split', to decide on merging or splitting

In case of splitting, the vector files named in line 2 and 3 are used to determine the number of basis functions in each fragment.

Output files

- AddOrb AB vectors as result of merging A and B
- INPORB_A vectors of fragment A after splitting
- INPORB_B vectors of fragment B after splitting

Input files

- INPORB Vectors to be split in fragments
- vector files defined in line 2 and 3 of the input can carry any name.

Input example:

```
Vectors obtained by merging fragment A and B
A.RasOrb
B.ScfOrb
baab
add
```

This merges the vectors of A and B, adopting the order: Occupied B, Occupied A (inactive + active), Virtual A, Virtual B, Merged vecotrs are written in AddOrb

5. ovlf - *merging the vectors of two overlapping fragments*

`$> ovlf x`	where x is the minimal singular value for considering an orbital to be present in both fragments, typically x = 0.2

Output files

- SUPERORB	AB vectors as result of merging A and B
- CORRORB.A	Corresponding inactive, active and virtual orbitals of fragment A
- CORRORB.B	Corresponding inactive, active and virtual orbitals of fragment B

Input files

- INPORB1	Vectors of fragment A
- INPORB2	Vectors of fragment B
- RUNFILE	RunFile of the full system
- ONEINT	1-el. integrals file of the full system
- GUESSORB	Any vector file of the full system (GssOrb generated by seward is good enough, only used to complete SUPERORB with some virtuals)

More details are given in the tutorial on the overlapping fragment method.

6. corr_shift - *post processing of the NOCI results, shifting diagonal matrix elements and additional sub-block diagonalizations*

Keyword	Description
Project	Prefix of the arx file for reading H and S
MEBF	Number of MEBFs (*n*), followed by *n* lines with the fragment functions that constitute the MEBF, same logic as in GronOR
Shift	Energy shift for each fragment function (in Eh)
Hamiltonian	*n* x *n* matrix with the Hamilton matrix elements
Overlap	*n* x *n* matrix with the MEBF overlaps
Block diagonalizations	Number of blocks (*b*) to be diagonalized, followed by *b* lines with the dimension of the block and the MEBFs included in that block
Select	Number of new MEBFs (x) to be selected for the calculation of 'dressed' coupling, followed by one line with x numbers listing the original MEBFs that should have maximal overlap with the new MEBFs for which the 'dressed' coupling will be calculated.

Hamiltonian and **Over**lap are ignored when **Proj**ect is defined, as this activates reading H and S from the arx file. In this example two extra blocks are defined: (i) S0S1, S1S0, D+D- and D-D+, and (ii) T1T1, D+D- and D-D+. From the 7 resulting eigenvectors, the new MEBFs, three are selected for the calculation of the 'dressed' coupling. The new MEBFs are selected to have maximal overlap with S0S1, S1S0 and T1T1.

No further input or output.

Input example:

```
MEBF
  6
 1   1   2   3   4   5
 6   7   6   8  10   9
Shift
  -1.5014 -1.5374 -1.5050 -1.4783 -1.5669  -1.5014 -1.5374 -1.5050 -1.4783 -1.5669
Hamiltonian
 -767.427       0.961      -0.961       2.440      31.723     -31.723
    0.961    -767.275      -0.089      -0.817     -22.429       3.245
   -0.961      -0.089    -767.275       0.817       3.245     -22.429
    2.440      -0.817       0.817    -767.241     -23.414      23.414
   31.723     -22.429       3.245     -23.414    -767.209       1.169
  -31.723       3.245     -22.429      23.414       1.169    -767.209
Overlap
 1.000    -0.001     0.001    -0.003    -0.041     0.041
-0.001     1.000     0.000     0.001     0.029    -0.004
 0.001     0.000     1.000    -0.001    -0.004     0.029
-0.003     0.001    -0.001     1.000     0.031    -0.031
-0.041     0.029    -0.004     0.031     1.000    -0.002
 0.041    -0.004     0.029    -0.031    -0.002     1.000
Block diagonalizations
 2
 4  2 3 5 6
 3  4 5 6
Select
 3
  2 3 4
```