(sum_rdn_)

Assign redundancy (rdn) to lesser-magnitude m|d in param pair.

if fPd : redundant_pairs = (I, L), (I, D), (I, M), (L, M), (D, M)
if not fPd : redundant_pairs = (I, L), (I, D), (I, M), (L, D), (D, M)

Example: if not fPd:

For (I, L) pair

if Idert.m > Ldert.m: rdn[1] += 1
if Ldert.m > Idert.m: rdn[0] += 1

Idert1	Idert2	Idert3	Idert4	Idert5	Idert6
rdn1 ↕	rdn2 ↕	rdn3 ↕	rdn4 ↕	rdn5 ↕	rdn6 ↕
Ldert1	Ldert2	Ldert3	Ldert4	Ldert5	Ldert6

rdn[0] rdn[1]

For (I, D) pair

if Idert.m > Ddert.m: rdn[1] += 1
if Ddert.m > Idert.m: rdn[0] += 1

Idert1	Idert2	Idert3	Idert4	Idert5	Idert6
rdn1 ↕	rdn2 ↕	rdn3 ↕	rdn4 ↕	rdn5 ↕	rdn6 ↕
Ddert1	Ddert2	Ddert3	Ddert4	Ddert5	Ddert6

rdn[0] rdn[1]

For (I, M) pair

if Idert.m > Mdert.m: rdn[1] += 1
if Mdert.m > Idert.m: rdn[0] += 1

Idert1	Idert2	Idert3	Idert4	Idert5	Idert6
rdn1 ↕	rdn2 ↕	rdn3 ↕	rdn4 ↕	rdn5 ↕	rdn6 ↕
Mdert1	Mdert2	Mdert3	Mdert4	Mdert5	Mdert6

rdn[0] rdn[1]

For (L, M) pair

if Ldert.m > Mdert.m: rdn[1] += 1
if Mdert.m > Ldert.m: rdn[0] += 1

Ldert1	Ldert2	Ldert3	Ldert4	Ldert5	Ldert6
rdn1 ↕	rdn2 ↕	rdn3 ↕	rdn4 ↕	rdn5 ↕	rdn6 ↕
Mdert1	Mdert2	Mdert3	Mdert4	Mdert5	Mdert6

rdn[0] rdn[1]

For (D, M) pair

if Ddert.m > Mdert.m: rdn[1] += 1
if Mdert.m > Ddert.m: rdn[0] += 1

Ddert1	Ddert2	Ddert3	Ddert4	Ddert5	Ddert6
rdn1 ↕	rdn2 ↕	rdn3 ↕	rdn4 ↕	rdn5 ↕	rdn6 ↕
Mdert1	Mdert2	Mdert3	Mdert4	Mdert5	Mdert6

rdn[0] rdn[1]

Sum each pair's rdn per element

Rdn_L1	Rdn_L2	Rdn_L3	Rdn_L4	Rdn_L5	Rdn_L6

Sum each pair's rdn per element

Rdn_I1	Rdn_I2	Rdn_I3	Rdn_I4	Rdn_I5	Rdn_I6

Sum each pair's rdn per element

Rdn_D1	Rdn_D2	Rdn_D3	Rdn_D4	Rdn_D5	Rdn_D6

Sum each pair's rdn per element

Rdn_M1	Rdn_M2	Rdn_M3	Rdn_M4	Rdn_M5	Rdn_M6