**/\* 4. The LH-diseases proper: the intersections of the disease-based LH DGP and the PC \*/**

/\* Which are the LH diseases (with the DGP) listed in OMIM, using as criterion of LH the condition ‘one phenotype id, more than one DGP id’? Second, which are the non-null LH-PC intersections? \*/

.open c:/sqlite/PC/pc.db

.mode tabs

.headers ON

/\* Identify the LH-diseases as the diseases annotated with more than one DGP (as entrez). \*/

CREATE TABLE tmp1\_omim\_LH(phen\_mim INTEGER, dgp\_count INTEGER);

INSERT INTO tmp1\_omim\_LH

SELECT DISTINCT \* FROM(

SELECT phen\_mim, COUNT(DISTINCT entrez\_id) AS dgp\_count FROM omim\_def

GROUP BY phen\_mim ORDER BY dgp\_count DESC);

DELETE FROM tmp1\_omim\_LH WHERE dgp\_count = 1;

/\* The unique phen\_mim/DGP pairs (with the DGP sorted alphabetically). \*/

CREATE TABLE tmp2\_omim\_LH(phen\_mim INTEGER, symbol TEXT);

INSERT INTO tmp2\_omim\_LH

SELECT DISTINCT \* FROM(

SELECT phen\_mim, symbol FROM omim\_def

ORDER BY phen\_mim, symbol);

/\* The unique phen\_mim/DGP pairs (with the DGP concatenated). \*/

CREATE TABLE tmp3\_omim\_LH(phen\_mim INTEGER, symbols TEXT);

INSERT INTO tmp3\_omim\_LH

SELECT phen\_mim, GROUP\_CONCAT(symbol,”;”) AS symbols FROM(SELECT DISTINCT \* FROM(

SELECT phen\_mim, symbol FROM tmp2\_omim\_LH ORDER BY phen\_mim, symbol))

GROUP BY phen\_mim;

/\* [Optional] If the names of the diseases is needed, proceed as follows. As the ‘phenotypes’ column in omim\_def is verbose and repetitive (different diseases being very similar to each other; for instance, colorectal cancer and colon cancer), use the ‘preferred term’ from the ‘mimTitles.txt’ file if a more concise output is desired. \*/

.import c:/sqlite/PC/inputs/mimTitles.txt tmp\_titles

CREATE TABLE titles(phen\_mim INTEGER, title TEXT);

INSERT INTO titles

SELECT DISTINCT \* FROM(

SELECT Mim\_Number, “Preferred\_Title; Symbol” FROM tmp\_titles);

/\* Assemble phenotype\_id, disease name, dgp count and DGP (concatenated). \*/

CREATE TABLE **omim\_LH**(phen\_mim INTEGER, phen\_title TEXT, dgp\_count INTEGER, symbols TEXT);

INSERT INTO omim\_LH

SELECT DISTINCT \* FROM(

SELECT a.phen\_mim, b.title, a.dgp\_count, c.symbols

FROM tmp1\_omim\_LH a

LEFT JOIN titles b ON a.phen\_mim = b.phen\_mim

JOIN tmp3\_omim\_LH c ON a.phen\_mim = c.phen\_mim);

.once c:/sqlite/PC/outputs/**omim\_LH.txt**

SELECT \* FROM omim\_LH;

DROP TABLE tmp\_titles;

DROP TABLE tmp1\_omim\_LH; DROP TABLE tmp2\_omim\_LH; DROP TABLE tmp3\_omim\_LH;

/\* **4bis. The intersection of disease-based LH and the PC** \*/

/\* How many cases of disease-based LH (from omim\_LH) are also shared membership of a PC? \*/

/\* Create the phen\_dgp doublets from the OMIM dataset. \*/

CREATE TABLE **phen\_dgp**(phen\_mim INTEGER, dgp INTEGER);

INSERT INTO phen\_dgp

SELECT DISTINCT \* FROM(

SELECT phen\_mim, entrez\_id FROM omim\_def);

/\* Assemble the phen\_dgp\_pc triplets starting from the phen\_dgp and pc\_dgp doublets. \*/

CREATE TABLE **phen\_dgp\_pc**(phen\_mim INTEGER, dgp INTEGER, go\_id TEXT);

INSERT INTO phen\_dgp\_pc

SELECT DISTINCT \* FROM(

SELECT a.phen\_mim, a.dgp, b.go\_id

FROM phen\_dgp a

JOIN pc\_dgp b ON a.dgp = b.dgp);

/\* Create table **output\_01\_phen** to search for pairs of different dgp (dgp\_i and dgp\_j) that cause the same disease (as phenotype\_mim) and that belong to the same GO-derived PC. Self-join the phen\_dgp\_pc table. \*/

CREATE TABLE tmp1\_output\_01\_phen(phen\_mim INTEGER, dgp\_i INTEGER, dgp\_j INTEGER, go\_id INTEGER);

INSERT INTO tmp1\_output\_01\_phen

SELECT DISTINCT \* FROM(

SELECT i.phen\_mim, i.dgp AS dgp\_i, j.dgp AS dgp\_j, i.go\_id

FROM phen\_dgp\_pc i

INNER JOIN phen\_dgp\_pc j

ON (i.phen\_mim = j.phen\_mim) AND (i.dgp < j.dgp) AND (i.go\_id = j.go\_id)

ORDER BY i.phen\_mim);

/\* Count the number of unique dgp shared by each disease/PC (phen\_mim/go\_id) pair. \*/

CREATE TABLE tmp2\_output\_01\_phen(phen\_mim INTEGER, go\_id INTEGER, dgp\_count INTEGER);

INSERT INTO tmp2\_output\_01\_phen

SELECT DISTINCT \* FROM(

SELECT phen\_mim, go\_id, COUNT(DISTINCT dgp) FROM ( ~~/\* tmp2\_output\_01\_phen \*/~~

SELECT DISTINCT \* FROM(

SELECT phen\_mim, dgp\_i AS dgp, go\_id FROM tmp1\_output\_01\_phen

UNION

SELECT phen\_mim, dgp\_j AS dgp, go\_id FROM tmp1\_output\_01\_phen)

)

GROUP BY phen\_mim, go\_id

ORDER BY COUNT(DISTINCT dgp) DESC);

CREATE TABLE tmp3\_output\_01\_phen(phen\_mim INTEGER, go\_id INTEGER, phen\_title TEXT, go\_name TEXT, dgp\_count INTEGER);

INSERT INTO tmp3\_output\_01\_phen

SELECT DISTINCT \* FROM(

SELECT a.phen\_mim, a.go\_id, b.title, c.go\_name, a.dgp\_count

FROM tmp2\_output\_01\_phen a

JOIN titles b ON a.phen\_mim = b.phen\_mim

JOIN pc c ON a.go\_id = c.go\_id);

/\* Calculate the phen\_mim length (i.e., the number of dgp they annotate). \*/

CREATE TABLE phen\_length(phen\_mim INTEGER, phen\_length REAL);

INSERT INTO phen\_length

SELECT DISTINCT \* FROM(

SELECT phen\_mim, COUNT (DISTINCT entrez\_id) FROM omim\_def

GROUP BY phen\_mim

ORDER BY COUNT(DISTINCT entrez\_id) DESC);

CREATE TABLE **output\_01\_phen**(phen\_mim INTEGER, go\_id INTEGER, phen\_title TEXT, go\_name TEXT, dgp\_count REAL, phen\_size REAL, pc\_size REAL);

INSERT INTO output\_01\_phen

SELECT DISTINCT \* FROM(

SELECT a.\*, b.phen\_length, c.pc\_length

FROM tmp3\_output\_01\_phen a

JOIN phen\_length b ON a.phen\_mim = b.phen\_mim

JOIN pc\_length c ON a.go\_id = c.go\_id);

ALTER TABLE output\_01\_phen ADD COLUMN Jacc FLOAT;

UPDATE output\_01\_phen SET Jacc = (dgp\_count/(phen\_size+pc\_size-dgp\_count));

UPDATE output\_01\_phen SET Jacc = ROUND(Jacc,2);

/\* Remove the PC that are too poorly specific. \*/

DELETE FROM output\_01\_phen WHERE(go\_id = ‘GO\_0043234’ OR go\_id = ‘GO\_0005667’ OR go\_id = ‘GO\_0043235’ OR go\_id = ‘GO\_0090575’);

DROP TABLE tmp1\_output\_01\_phen; DROP TABLE tmp2\_output\_01\_phen;

DROP TABLE tmp3\_output\_01\_phen;

.once c:/sqlite/PC/outputs/output\_01\_phen.txt

SELECT \* FROM output\_01\_phen;