nature portfolio

https://doi.org/10.1038/s41596-021-00597-z

Supplementary information

A practical guide to large-scale docking

In the format provided by the authors and unedited

Supplementary Information for A Practical Guide to Large Scale Docking

Authors: Brian J. Bender[1,4], Stefan Gahbauer[1,4], Andreas Luttens[2], Jiankun Lyu[1], Chase M. Webb[1], Reed M. Stein[1], Elissa A. Fink[1], Trent E. Balius[3], Jens Carlsson[2], John J. Irwin[1], Brian K. Shoichet[1,5]

Affiliations:
[1]Department of Pharmaceutical Chemistry, University of California-San Francisco, San Francisco, CA, USA 94158
[2]Science for Life Laboratory, Department of Cell and Molecular Biology, Uppsala University, SE-751 24, Uppsala, Sweden
[3]NCI RAS Initiative, Cancer Research Technology Program, Frederick National Laboratory for Cancer Research, Leidos Biomedical Research, Inc., Frederick, MD 21702
[4]These authors contributed equally

[5]Correspondence: shoichet@cgl.ucsf.edu

Table of Contents

INDOCK Guide

Example INDOCK file with explanation of the options and guidance on which can be modified.

```
DOCK 3.7 parameter
#######################################################
# NOTE: split_database_index is reserved to specify a list of files
# defaults for large scale docking.
ligand_atom_file        split_database_index
#######################################################
#                       OUTPUT
output_file_prefix      test.
#######################################################
#                       MATCHING
match_method                    2       #  1 matches up to the distance_tolerance below, ignoring match_goal,
                                        ###  step, maximum, etc.
                                        #  2 uses the adaptive sampling that attempts to get a number of
                                        ###  match_goal orientational samples
distance_tolerance              0.05    #  starting distance tolerance
match_goal                      1000    #  desired number of orientational samples to get before quitting using
                                        ###  match_method = 2; increasing to 2500 or 5000 helpful for large and
                                        ###  flexible ligands, but will slow calculations
distance_step                   0.05    #  increment from distance_tolerance until max or match_goal is reached
distance_maximum                0.5     #  biggest tolerance that will be used to attempt to get match_goal
                                        ###  orientational samples
timeout                         10.0    #  number of seconds before quitting on any given ligand
nodes_maximum                   4       #  max number of points for which all distances must be within the
                                        ###  tolerance, 3 possible, 4 suggested
nodes_minimum                   4       #  min number of points for which all distances must be within the
                                        ### tolerance, 4 suggested, 3 possible
bump_maximum                    10.0    #  van der Waals score in kcal/mol for any part of the molecule to get
                                        ### before further examination stopped; can be increased
bump_rigid                      10.0    #  van der Waals score in kcal/mol for the rigid component of the ligand
                                        ### molecule allowed before the pose discarded; can be increased for
                                        ### large, floppy ligands
mol2_score_maximum              -10.0   #  poses with scores above this value will not be saved; can be increased
                                        ### in retrospective calculations, should be decreased for large-scale
                                        ### prospective screens to save on disk space
#######################################################
#                       COLORING
chemical_matching               no      #  default to off, can use chemical matching from DOCK3.6
                                        ###  not recommended
case_sensitive                  no      #  case sensitivity for chemical_matching groups
```

```
##########################################################
#                SEARCH MODE
atom_minimum            4       #  minimum number of atoms in ligand for it to be scored
atom_maximum            25      #  maximum number of atoms in ligand for it to be scored, increase to 100
number_save             1       #  how many poses to save; any number can be saved but disk space
                                ### is a limiting factor
number_write            1       #  how many poses to write out
flush_int               100
#molecules_maximum      100000 # how many molecules will be searched before quitting.
                                ### Leave commented out
check_clashes           yes     # check for internal clashes; clashes are pre-computed during db2
                                ### generation and stored in db2 file in the first line for each set (S).
                                ### If check_clash equals yes then all sets with any number of clashes
                                ### (clash number > 0) will not be docked
do_premax               no
do_clusters             no
##########################################################
#                SCORING
ligand_desolvation      volume          # use GB desolvation scoring, other options are full or none
#vdw_maximum            1.0e10          # maximum vdw score possible, prevents overflow
ligand_desolv_scale     1.0     # scaling factors to be applied to scores, leave unchanged
electrostatic_scale     1.0     # scaling factors to be applied to scores, leave unchanged
vdw_scale               1.0     # scaling factors to be applied to scores, leave unchanged
internal_scale          0.0     # leave unchanged, scales an internal focusing term
per_atom_scores         no      # change to yes if per-atom scoring breakdown is desired
                                ### note that this doubles output size
##########################################################
#                DOCKovalent     # only to be changed for covalent docking, not described here
dockovalent             no
bond_len                1.8
bond_ang1               109.5
bond_ang2               109.5
len_range               0.0
len_step                0.1
ang1_range              10.0
ang2_range              10.0
ang1_step               2.5
ang2_step               2.5
##########################################################
#             MINIMIZATION        # perform ligand minimization, recommended to leave unchanged
minimize                yes
sim_itmax               500
sim_trnstep             0.2
sim_rotstep             5.0
sim_need_to_restart     1.0
sim_cnvrge              0.1
min_cut                 1.0e15
iseed                   777
```

```
##########################################################
# INPUT FILES / THINGS THAT CHANGE          #### values below defined by unique receptor
receptor_sphere_file        ../dockfiles/matching_spheres.sph
vdw_parameter_file          ../dockfiles/vdw.parms.amb.mindock
delphi_nsize                71              # binding pocket-specifc, can be check with phi_to_dx.py
flexible_receptor           no
total_receptors             1
############### grids/data for one receptor
rec_number                  1
rec_group                   1
rec_group_option            1
solvmap_file                ../dockfiles/ligand.desolv.heavy
hydrogen_solvmap_file       ../dockfiles/ligand.desolv.hydrogen
delphi_file                 ../dockfiles/trim.electrostatics.phi
chemgrid_file               ../dockfiles/vdw.vdw
bumpmap_file                ../dockfiles/vdw.bmp
############### end of INDOCK
```

Blastermaster Guide

Steps of the blastermaster.py script:

1.	reduce	Protonates protein target structure
2.	filter	Selects binding site residues based on xtal-lig.pdb position
3.	dms	Generates molecular surface
4.	sphgen	Generates spheres describining the negative image of the binding pocket
5.	thin_spheres.py	Generates boundary layer spheres: low dielectric and ligand desolvation
6.	close_sph.py	Selects boundary spheres according to distance from xtal-lig.pdb
7.	pdbtosph	Converts xtal-lig.pdb to spheres
8.	makespheres1.cli.pl	Selects low dielectric spheres from xtal-lig.match.sph and all_spheres.sph
9.	doshowsph.csh	Converts .sph files to .pdb files and add spheres to rec.crg.pdb
10.	makespheres3.cli.pl	Generates matching spheres
11.	makebox	Builds box for DOCK scoring grid generation
12.	qnifft	Makes electrostatic grid (phi-map) based on the Poisson-Boltzmann equation
13.	phiTrim.py	Trims phi-map to box
14.	chemgrid	Generates van der Waals scoring grid
15.	solvmap	Generates ligand desolvation grid
16.	blasterIndock.py	Generates INDOCK file

The following outlines the default values and command lines used in each step of blastermaster. Although not recommended for general use, specific steps can be rerun for user defined modifications.

1. reduce

Purpose: Protonate protein target

Inputs:

- rec.pdb
- $DOCKBASE/proteins/default/reduce_wwPDB_het_dict.txt

Outputs:

- rec.crg.pdb.fullh
- addh.log
- rec.crg.pdb
- rec.crg.pdb.polarH

Command line:

$DOCKBASE/proteins/Reduce/reduce -db reduce_wwPDB_het_dict.txt -OH -HIS -ALLALT -NOROTNH3 -Keep -METALBump1.5 -NONMETALBump-5.0 rec.pdb > rec.crg.pdb.fullh

Associated flags:

- --addNOhydrogensflag
- --addhProgram
- --addhDict
- --addhOptions
- --addhFirst
- --addhFirstOptions
- --chargedPdbOutput

Blastermaster function:

- blasterAddHydorgens.py
- pdb.py

2. filter

Purpose: Select binding site residues based on xtal-lig.pdb position

Inputs:

- rec.pdb
- xtal-lig.pdb
- $DOCKBASE/proteins/defaults/filt.params

Outputs:

- filter.log
- rec.site

Command line:

$DOCKBASE/proteins/filt/bin/filt < $DOCKBASE/proteins/defaults/filt.params > filter.log

Associated flags:

- --filterProgram
- --filterLog
- --bindsiteResidues

Blastermaster function:

- blasterBindingSiteResidues.py

3. dms

Purpose: Generate molecular surface. The molecular surface is required for sphere generation, e.g. matching spheres, lowdielectric spheres, boundary spheres

Inputs:

- rec.crg.pdb.dms
- rec.site.dms
- $DOCKBASE/proteins/defaults/radii

Outputs:

- rec.ms
- dms.log
- rec.ts.ms
- dms.ts.log

Command lines:

$DOCKBASE/proteins/dms/bin/dms rec.crg.pdb.dms -a -d 1.0 -i rec.site.dms -g dms.log -p -n -o rec.ms

$DOCKBASE/proteins/dms/bin/dms rec.crg.pdb.dms -a -d 1.0 -i rec.site.dms -g dms.ts.log -p -n -o rec.ts.ms

Associated flags (defaults):

- --msProgram
- --msOutput
- --mstsOutput
- --msLog
- --mstsLog
- --msDensity (1.0)
- --mstsDensity (1.0)
- --msRadiusFile

Blastermaster function:

- blasterMolSurf.py

Comments:

rec.crg.pdb.dms and rec.site.dms are rec.crg.pdb and rec.site without waters, respectively.

If thin spheres (ts) are used (--useThinSphEleflag –useThinSphLdsflag) dms is run twice.

4. sphgen

Purpose: Generate spheres describing negative image of binding pocket.

Inputs:

- /working/INSPH
- (rec.ms)

Outputs:

- OUTSPH
- all_spheres.sph

Command line:

$DOCKBASE/proteins/sphgen/bin/sphgen

Associated flags:

- --sphgenProgram
- --sphgenOutput

Blastermaster function:

- blasterSphgen.py

5. thin_spheres.py

Purpose: Generates boundary layer spheres: low dielectric and ligand desolvation (runs twice)

Inputs:

- rec.ts.ms

Outputs:

- low_die_thinspheres.sph

- lig_des_thinspheres.sph

Command lines:

$DOCKBASE/proteins/thinspheres/thin_spheres.py -i rec.ts.ms -o low_die_thinspheres.sph -d 1.0 -s 1.0

$DOCKBASE/proteins/thinspheres/thin_spheres.py -i rec.ts.ms -o lig_des_thinspheres.sph -d 1.0 -s 1.0

Associated flags (defaults):

1. --useThinSphEleflag
2. --useThinSphLdsflag
3. --thinSphProgram
4. --sphOutput_ele
5. --ts_dist_ele (1.0)
6. --ts_radius_ele (1.0)
7. --sphOutput_lds
8. --ts_dist_lds (1.0)
9. --ts_radius_lds (1.0)

Blastermaster function:

- blasterThinSpheres.py

Notes:

“Thin spheres” is a historic name for dielectric boundary surface spheres as they add a ”thin” low-dielectric layer onto the protein surface.

6. close_sph.py

Purpose: Selects boundary spheres according to distance from xtal-lig.pdb (runs twice)

Inputs:

- low_die_thinspheres.sph
- lig_des_thinspheres.sph

Outputs:

- low_die_thinspheres.sph.close
- low_die_thinspheres.sph.close.log
- lig_des_thinspheres.sph.close
- lig_des_thinspheres.sph.close.log

Command lines:

python $DOCKBASE/proteins/thinspheres/close_sph.py low_die_thinspheres.sph xtal-lig.pdb low_die_thinspheres.sph.close 2.0 1.0

python $DOCKBASE/proteins/thinspheres/close_sph.py low_des_thinspheres.sph xtal-lig.pdb low_des_thinspheres.sph.close 2.0 1.0

Associated flags (defaults):

- --closeSphProgram
- --ts_radius_ele (1.0)
- --ts_radius_lds (1.0)

- --ts_dist_to_lig (2.0)

Blastermaster function:

- blasterThinSpheres.py

7. pdbtosph

Purpose: Convert xtal-lig.pdb to spheres

Input:

- xtal-lig.pdb

Output:

- xtal-lig.match.sph

Command line:

$DOCKBASE/proteins/pdbtosph/bin/pdbtosph xtal-lig.pdb xtal-lig.match.sph

Associated flags:

- --pdbsphProgram
- --pdbsphOutput

Blastermaster function:

- blasterPdbsph.py

8. makespheres1.cli.pl

Purpose: Selects low dielectric spheres from xtal-lig.match.sph and all_spheres.sph if thin spheres flag is not used.

Inputs:

- xtal-lig.match.sph
- all_spheres.sph
- rec.crg.pdb

Outputs:

- lowdielectric.sph
- lowdielectric_spheres.log

Command line:

$DOCKBASE/proteins/makespheres1/makespheres1.cli.pl xtal-lig.match.sph all_spheres.sph rec.crg.pdb lowdielectric.sph 25 >& lowdielectric.spheres.log

- Associated flags (defaults):
- --lowdielectricSpheresProgram
- --lowdielectricSpheresOutput
- --lowdielectricSpheresLog
- --minLowdielectricSpheres (25)

Blastermaster function:

- blasterLowDielectricSpheres.py

9. doshowsph.csh

Purpose: Converts .sph files to .pdb files and add spheres to rec.crg.pdb.

Inputs:

- lowdielectric.sph
- low_die_thinspheres.sph.close
- lig_des_thinspheres.sph.close
- rec.crg.pdb

Outputs:

- lowdielectric.sph.pdb
- lowdielectric.sph.pdb.log
- low_die_thinspheres.sph.close.pdb
- low_die_thinspheres.sph.close.pdb.log
- lig_des_thinspheres.sph.close.pdb
- lig_des_thinspheres.sph.close.pdb.log
- receptor.crg.lowdielectric.pdb

Command lines:

$DOCKBASE/proteins/showsphere/doshowsph.csh lowdielectric.sph 1 lowdielectric.sph.pdb >& lowdielectric.sph.pdb.log

$DOCKBASE/proteins/showsphere/doshowsph.csh low_die_thinspheres.sph.close 1 low_die_thinspheres.sph.close.pdb >& low_die_thinspheres.sph.close.log

$DOCKBASE/proteins/showsphere/doshowsph.csh lig_des_thinspheres.sph.close 1 lig_des_thinspheres.sph.close.pdb >& lig_des_thinspheres.sph.close.log

cat rec.crg.pdb lowdielectric.sph.pdb > receptor.crg.lowdielectric.pdb

Associated flags:

- --sphtopdbProgram
- --lowdielectricPdbOutput
- --lowdielectricPdbLog
- --receptorLowdielectricOutput

Blastermaster function:

- blasterSphtopdb.py
- blasterCat.py

10. makespheres3.cli.pl

Purpose: Generates matching spheres.

Inputs:

- xtal-lig.match.sph
- all_spheres.sph
- rec.crg.pdb

Outputs:

- matching_spheres.sph (dockfiles)
- matching_spheres.log

Command line:

$DOCKBASE/proteins/makespheres3/makespheres3.cli.pl 1.5 0.8 45 xtal-lig.match.sph all_spheres.sph rec.crg.pdb matching_spheres.sph >& matching_spheres.log

Associated flags (defaults):

- --matchingSpheresProgram
- --matchingSPheresOutput
- --matchingSpheresLog
- --matchingSpheresDist1 (1.5)
- --matchingSpheresDist2 (0.8)
- --matchingSpheresMax (45)

Blastermaster function:

- blasterMatchingSpheres.py

11. makebox

Purpose: Builds box for DOCK scoring grid preparation

Inputs:

- xtal-lig.match.sph
- rec.crg.pdb

Outputs:

- box
- makebox.log

Command line:

$DOCKBASE/proteins/makebox/makebox.smallokay.pl xtal-lig.match.sph rec.crg.pdb box 10.0 >& makebox.log

Associated flags (defaults):

- --boxProgram
- --boxOutput
- --boxLog
- --box_margin (10.0)

Blastermaster function:

- blasterBox.py

12. qnifft

Purpose: Produces electrostatic grid (phi-map) by numeric solution of the Poisson Boltzmann equation

Inputs:

- $DOCKBASE/proteins/defaults/qnifft.parm
- (receptor.crg.lowdielectric.pdb, amb.crg.oxt, vdw.siz)

Outputs:

- qnifft.atm
- qnifft.electrostatics.phi
- qnifft.eps
- qnifft.log
- qnifft_sas.usr

Command line:

$DOCKBASE/proteins/qnifft/bin/qnifft22_193_pgf_32 qnifft.parm >& qnifft.log

Associated flags (defaults):

- --qnifftProgram
- --qnifftOut
- --qnifftPdbOut
- --qnifftLog
- --qnifftGrid (193)
- --chargeFile (amb.crg.oxt)
- --radiusFile (vdw.siz)

Blastermaster function:

- blasterQnifft.py

13. phiTrim.py

Purpose: Trims phi-map to box

Inputs:

- qnifft.electrostatics.phi
- box

Output:

- trim.electrostatics.phi (dockfiles)

Command line:

$DOCKBASE/proteins/blastermaster/phiTrim.py qnifft.electrostatics.phi box trim.electrostatics.phi

Associated flag:

- --qnifftTrimOut

Blastermaster functions:

- phiTrim.py

- phi.py
- box.py
- combinatorics.py

14. chemgrid

Purpose: Generates van der Waals scoring grids.

Inputs:

- /working/INCHEM
- (rec.crg.pdb, $DOCKBASE/proteins/defaults/prot.table.ambcrg.ambH, $DOCKBASE/proteins/defaults/vdw.parms.amb.mindock, box)

Outputs:

- vdw.bmp (dockfiles)
- vdw.esp
- vdw.log
- vdw.vdw (dockfiles)

Command line:

$DOCKBASE/proteins/chemgrid/bin/chemgrid >& vdw.log

Associated flags (defaults):

- --vdwProgram
- --vdwLog
- --vdwprottable
- --vdwparameters
- --vdw
- --vdwgridspacing (0.2)

Blastermaster function:

- blasterChemGrid.py

15. solvmap

Purpose: Generates ligand desolvation scoring grids. Solvmap runs twice, first for ligand heavy atoms, then for ligand hydrogen atoms.

Inputs:

- rec.crg.lds.pdb (copied from rec.crg.pdb; if thin ligand desolvation spheres are present, they are added)
- box
- /working/heavy/INSEV
- /working/hydrogen/INSEV

Outputs:

- /*/OUTSEV
- /*/solvmap.log
- /*/dielec_sev.box

- /*/solv_sev.box
- /*/solv_sev.plt
- *heavy; hydrogen
- /heavy/ligand.desolv.heavy (dockfiles)
- /hydrogen/ligand.desolv.hydrogen (dockfiles)

Command line:

$DOCKBASE/proteins/solvmap/bin/solvmap >& solvmap.log

Associated flags (defaults):

- --solvmapProgram
- --solvmapLog
- --solvmapHydrogenRadius (1.0)
- --solvmapHeavyRadius (1.8)
- --solvmapProbeRadius (1.4)
- --solvmapHydrogenName
- --solvmapHeavyName

Blastermaster function:

- blasterSolvmap.py

16. blasterIndock.py

Purpose: Generates INDOCK file

Output:

- INDOCK

Blastermaster function:

- blasterIndock.py
- Boltzmann.py