A Buildup-based Error Minimization Method with Application to Protein Structure Determination

Zhenli Sheng (盛镇醴)

email: szl@lsec.cc.ac.cn

Institute of Computational Mathematics and Scientific/Engineering Computing,
Chinese Academy of Sciences

joint work with Ya-xiang Yuan

July 31th, 2013

ICCOPT2013-Lisbon

Outline

Outline

Distance Geometry Problem

Find the coordinate vectors $x_1, x_2, \dots, x_n$ that satisfy several given distances among them. Mathematically, this problem can be stated as following,

Find $x_1, x_2, \dots, x_n$, such that

$$\|x_i - x_j\| = d_{ij}, \quad (i,j) \in S.$$

or

$$l_{ij} \leq \|x_i - x_j\| \leq u_{ij}, \quad (i,j) \in S.$$

- The given data may have noise.
- This problem can be formulated as a global optimization problem.
- It has many applications.

Global optimization: traditional error functions

- stress function

$$Stress(x_1, x_2, \ldots, x_n) = \sum_{(i,j)\in S} (\|x_i - x_j\| - d_{i,j})^2,$$

- smoothed stress function

$$SStress(x_1, x_2, \ldots, x_n) = \sum_{(i,j)\in S} (\|x_i - x_j\|^2 - d_{i,j}^2)^2,$$

- generalized stress function

$$GStress(x_1, x_2, \ldots, x_n) = \sum_{(i,j)\in S} \min{}^2\{\frac{\|x_i - x_j\|^2 - l_{i,j}^2}{l_{i,j}^2}, 0\} + \max{}^2\{\frac{\|x_i - x_j\|^2 - u_{i,j}^2}{u_{i,j}^2}, 0\}.$$

It has too many local minimizers, thus is very difficult to find the global solution.

Application I: Graph Realization

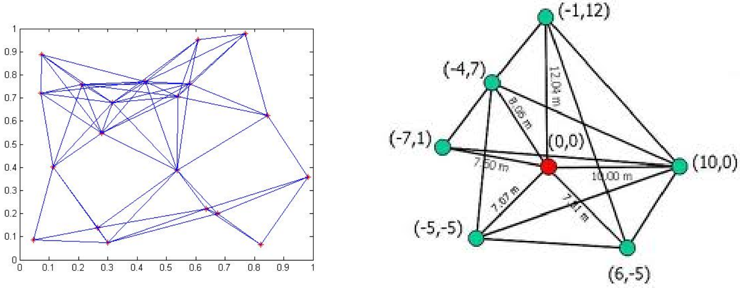


Figure 1 : Graph Realization in 2D

Given a graph G=(V,E), each edge has a weight.

Application II: Protein Structure Determination

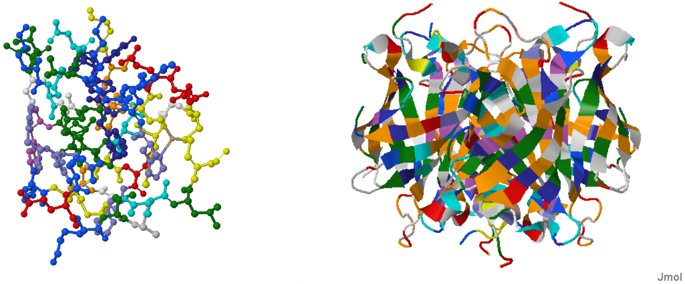


Figure 2 : Two proteins: 1PTQ and 1HQQ, in different display ways

- 3D problem.
- Measure distances by NMR or X ray crystallography.
- Many properties of proteins rely on structure.

Application III: Sensor Network Localization

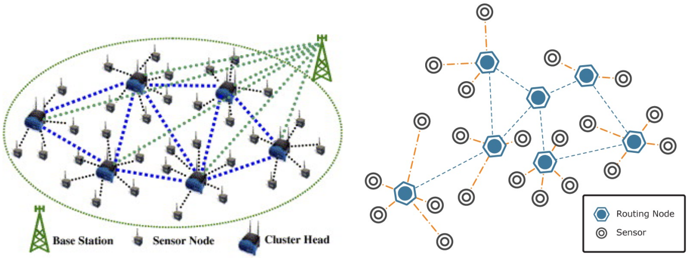


Figure 3 : Illustration of wireless sensor networks: anchor and sensor

- 2D problem.
- There are some anchors: locations are known.

Outline

2. **Related works review**
   - Matrix decomposition method
   - Buildup method

Matrix Decomposition Method

Matrix decomposition method [Blumenthal 1953, Torgerson 1958] works for Distance Geometry problem with full set of exact distances.

Given a full set of distances, $d_{ij} = \|x_i - x_j\|, \quad i, j = 1, 2, \dots, n.$

- Set $x_n = (0, 0, 0)^{\mathrm{T}}$, thus $d_{in} = \|x_i\|$, we have

$$\begin{aligned} d_{ij}^2 &= \|x_i - x_j\|^2 \\ &= \|x_i\|^2 - 2x_i^{\mathrm{T}} x_j + \|x_j\|^2 \\ &= d_{in}^2 - 2x_i^{\mathrm{T}} x_j + d_{jn}^2, \qquad i, j = 1, 2, \dots, n-1 \end{aligned} \tag{1}$$

- Define $X = (x_1, x_2, \dots, x_n)^{\mathrm{T}}$ and $D = \{(d_{in}^2 - d_{ij}^2 + d_{jn}^2)/2 : i, j = 1, 2, \dots, n-1\}$, (1) $\Rightarrow XX^{\mathrm{T}} = D$.

- Let $D = U\Sigma U^{\mathrm{T}}$, $V = U(:, 1:3)$ and $\Lambda = \Sigma(1:3, 1:3)$. Then $X = V\Lambda^{1/2}$ is the best rank-3 approximation. [Eckart-Young 1936]

Buildup Method

Algorithm 1: Buildup Method for Distance Geometry Problem with sparse data

Given: distance matrix D, i.e. $d_{ij}, (i,j) \in S$.

Step 1: Find a clique of four points and determine their coordinates.

Step 2: Choose a point to be added, apply liner or nonlinear least square to locate the point.

Step 3: Repeat Step 2 until all points are determined or no more points can be determined.

♠ The algorithm bases on a simple observation: in $\mathbb{R}^3$, four distances can uniquely determine a point.

♠ It was first proposed by [Dong-Wu 2002]. We will specify details of each step later.

Determine first four points

- Find a clique of four points: greedy search
- How to determine their coordinates?

  Denote $x_i = (u_i, v_i, w_i)^{\mathrm{T}}$, set $x_1 = (0,0,0)^{\mathrm{T}}$ and $x_2 = (d_{12}, 0, 0)^{\mathrm{T}}$.

$$\begin{cases} u_3^2 + v_3^2 = d_{31}^2 \\ (u_3 - u_2)^2 + v_3^2 = d_{32}^2 \end{cases} \Rightarrow \begin{cases} u_3 = (d_{31}^2 - d_{32}^2 + u_2^2)/(2u_2) \\ v_3 = \sqrt{d_{31}^2 - u_3^2}, \quad w_3 = 0 \end{cases}$$

$$\begin{cases} u_4^2 + v_4^2 + w_4^2 = d_{41}^2 \\ (u_4 - u_2)^2 + v_4^2 + w_4^2 = d_{42}^2 \\ (u_4 - u_3)^2 + (v_4 - v_3)^2 + w_4^2 = d_{43}^2 \end{cases} \Rightarrow \begin{cases} u_4 = (d_{41}^2 - d_{42}^2 + u_2^2)/(2u_2) \\ v_4 = \dots \\ w_4 = \sqrt{d_{41}^2 - u_4^2 - v_4^2} \end{cases}$$

$$v_4 = (d_{42}^2 - d_{43}^2 + (u_4 - u_2)^2 + (u_4 - u_3)^2 + v_3^2)/(2v_3)$$

- In the noisy case, the "sqrt" part may cause problems.

Determine the chosen point

Suppose x_j (to be determined) has l known distances with $x_i, x_2, \ldots, x_l$ (has been determined).

- linear least square [Wu-Wu 2007]:
  $d_{ij}^2 = \|x_i\|^2 - 2x_i^{\mathrm{T}} x_j + \|x_j\|^2, (i = 1, 2, \ldots, l). \Rightarrow Ax_j = b$, where

$$A = 2 \begin{pmatrix} x_{11} - x_{21} & x_{12} - x_{22} & x_{13} - x_{23} \\ x_{21} - x_{31} & x_{22} - x_{32} & x_{23} - x_{33} \\ \vdots & \vdots & \vdots \\ x_{l-1,1} - x_{l1} & x_{l-1,2} - x_{l2} & x_{l-1,3} - x_{l3} \end{pmatrix},$$

$$b = \begin{pmatrix} (\|x_1\|^2 - \|x_2\|^2) - (d_{1j}^2 - d_{2j}^2) \\ (\|x_2\|^2 - \|x_3\|^2) - (d_{2j}^2 - d_{3j}^2) \\ \vdots \\ (\|x_{l-1}\|^2 - \|x_l\|^2) - (d_{l-1,j}^2 - d_{lj}^2) \end{pmatrix}$$

♠ In the noisy case, solve $\min_{x_j} \|Ax_j - b\|_2$.

Determine the chosen point

Suppose x_j (to be determined) has l known distances with $x_1, x_2, \ldots, x_l$ (has been determined).

- nonlinear least square [Sit-Wu-Yuan 2009]:
  1. Calculate missing distances among these l points.
  2. All the distances among $l+1$ points are known, we apply matrix decomposition method to solve it.
  3. Move these points back to the original reference system, using $x_i, x_2, \ldots, x_l$ as common points.

  ♣ In this way, we determine the new point, as well as re-determine the l points, which can be viewed as adjustment.

Remarks about Buildup method

- Buildup method is almost perfect when the given distances are exact, because it is fast and accurate.
- Due to accumulation of round error, the result may damage if the number of the points is large (more than several thousands).
- Buildup method with nonlinear technique is more stable. However, it still can tolerate very small noise, usually no bigger than 0.01%.

Outline

3. Our algorithm

Motivation

Two observations:

- It is almost impossible to find the global minimizer or a good local minimizer of any error function, from a randomly chosen initial point.
- Buildup method is extremely fast, which is its most charming feature, but it cannot tolerate large noises.
- Many existing methods can be viewed as two processes:
  1) Find a good initial point:
     Embedding Method [Crippen-Havel 1988] trys to estimate the missing data, and then apply matrix decomposition method to obtain a solution.
     SDP Method [Biswas-Toh-Ye 2007] solves a relaxed SDP problem to produce an initial point.
  2) Minimize an error function to adjust the positions.
     remark: these methods don't talk too much about this procedure, but actually it is very crucial to their methods.

Our basic idea:

- Incorporate Buildup Method and error function minimization.

Our algorithm

Algorithm 2: Buildup-based Error Minimization Method for Distance Geometry Problem with sparse and noisy distances

Given: distance matrix D, i.e. $d_{ij}, (i, j) \in S$.

Step 1: Find a clique of four points and determine their coordinates.

Step 2: Choose a point to be added, apply nonlinear least square to roughly locate the point.

Step 3: Apply error minimization to these $l + 1$ points.

Step 4: Repeat Step 2-3 until all points are determined or no more points can be determined.

Step 5: Apply error minimization to all the determined points.

How to choose a new point?

Let $\mathcal{Y}$ be index set whose position bas been determined and $\mathcal{N} = \{1, 2, \ldots, n\} \backslash \mathcal{Y}$, we define

$$deg(i) = \sharp\{d_{ik} \neq 0 \mid i \in \mathcal{N}, k \in \mathcal{Y}\}$$

Then, in Step 2 we choose point j, such that

$$j = \arg\min_{i}\{\sum_{k \in \mathcal{Y}} d_{ik} \mid deg(i) = \max_{i \in \mathcal{N}} deg(i)\} \tag{2}$$

(Animation of the computational order)

New error function

Existing error functions focus on measuring the (relative) difference error, we propose the following two error function to measure quotient error.

$$f(x_1, x_2, \ldots, x_n) = \sum_{(i,j)\in S} h\left(\frac{\|x_i - x_j\|}{d_{ij}}\right)$$

where

$$h(x) = \begin{cases} \frac{1}{2}(x-1)^2, & \text{if } x \geq 1, \\ x - 1 - log(x), & \text{if } 0 < x < 1. \end{cases}$$

or

$$h(x) = \quad x - 1 - log(x), \quad x > 0.$$

Theorem 3.1

$h(x)$ is twice continuously differentiable in $(0, +\infty)$, $\lim_{x\to 0} h(x) = \infty$, $\lim_{x\to\infty} h(x) = \infty$, and it achieves its minimum 0 at 1.

Why this function?

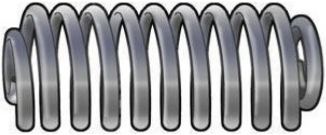


Figure 4 : Hooke's law models the properties of springs for small changes in length

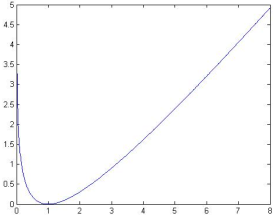
- "Force" function:

$$F = \begin{cases} x - 1, & x \geq 1, \\ 1 - \frac{1}{x}, & x < 1. \end{cases}$$

  then the energy function is h(x).

- Another proposal:

  $h(x) = x - (1 + ln(x))$



Properties of new error function

Theorem 3.2

Suppose

$$
\begin{aligned}
f(x) &= \sum_{(i,j)\in S} h\left(\frac{\|x_i - x_j\|}{d_{ij}}\right) \\
&= \sum_{(i,j)\in S} \begin{cases} \frac{1}{2}\left(\frac{\|x_i - x_j\|}{d_{ij}} - 1\right)^2, & \mathit{if}\ \|x_i - x_j\| \geq d_{ij}, \\ \frac{\|x_i - x_j\|}{d_{ij}} - 1 - log\left(\frac{\|x_i - x_j\|}{d_{ij}}\right), & \mathit{otherwise} \end{cases}
\end{aligned}
$$

let $N(i) = \{j : d_{ij} \neq 0\}$ *denote the neighbours of point* i*, and* $x = (x_1; x_2; \ldots; x_n)$ *is a column vector in* $\mathbb{R}^{3n}$*, then we have*

$$
\frac{\partial f}{\partial x_i} = \sum_{j \in N(i)} \begin{cases} \left(\frac{1}{d_{ij}^2} - \frac{1}{d_{ij}\|x_i - x_j\|}\right)(x_i - x_j), & \mathit{if}\ \|x_i - x_j\| \geq d_{ij}, \\ \left(\frac{1}{d_{ij}\|x_i - x_j\|} - \frac{1}{\|x_i - x_j\|^2}\right)(x_i - x_j), & \mathit{otherwise} \end{cases},
$$

to be continued...

Properties of new error function (cont'd)

Theorem 3.3

and for $p, q = 1, 2, 3,$

$$\frac{\partial^2 f}{\partial x_{ip} \partial x_{jq}} = \begin{cases} \sum_{j \in N(i)} \omega(x), & \textit{if } i = j, \\ -\omega(x), & \textit{if } i \neq j, \\ 0, & \textit{otherwise} \end{cases}$$

where

$$\omega(x) = \begin{cases} \frac{1}{d_{ij}^2} - \frac{1}{d_{ij} \|x_i - x_j\|} + \frac{(x_{ip} - x_{jq})^2}{d_{ij} \|x_i - x_j\|^3}, & \textit{if } \|x_i - x_j\| \geq d_{ij}, \\ \frac{1}{d_{ij} \|x_i - x_j\|} - \frac{1}{\|x_i - x_j\|^2} - \frac{(x_{ip} - x_{jq})^2}{d_{ij} \|x_i - x_j\|^3} + \frac{(x_{ip} - x_{jq})^2}{\|x_i - x_j\|^4}, & \textit{otherwise.} \end{cases}$$

Our algorithm

Algorithm 3: Buildup-based Error Minimization Method for Distance Geometry Problem with sparse and noisy distances

Given: distance matrix D, i.e. $d_{ij}, (i, j) \in S$.

Step 1: Find a clique of four points and determine their coordinates.

Step 2: Choose a point to be added, apply nonlinear least square to roughly locate the point.

Step 3: Apply error minimization to these $l + 1$ points.

Step 4: Repeat Step 2-3 until all points are determined or no more points can be determined.

Step 5: Apply error minimization to all the determined points.

Algorithm details

- In Step 3, we solve the following subproblem:

$$\min_{x_j, x_k \in N(i)} f(x_j, x_k)$$

  where f is an error function.

- The subproblems are relatively very small, thus can be solved very quickly.
- In Step 5, we solve

$$\min_{x_1, x_2, \ldots, x_n} f(x_1, x_2, \ldots, x_n)$$

- Since we can obtain very good starting point, we only exploit gradient method — line search BB method to solve these problems.

Buildup VS. our algorithm

- We re-use the same distance information (but only the given distances, not include calculated ones).
- In Buildup method equipped with nonlinear least square, if distances are noisy, then we have

$$\|x_i - x_j\| = d_{ij} + e_{ij},$$

where e_{ij} denotes noise.

$$\Rightarrow 2x_i^{\mathrm{T}} x_j = d_{in}^2 - d_{ij}^2 + d_{jn}^2 - 2d_{ij}e_{ij} - e_{ij}^2$$

- So, our conclusion is:

When the noises are large, $(d_{ij}e_{ij} + e_{ij}^2/2)$ should not be ignored, namely, solving $X^{\mathrm{T}}X = D$ can not give a good solution to the original problem. But it can serve as a warm starting point to minimize an error function to obtain a better solution.

Outline

4. Numerical experiments

Numerical experiments

We simulate on real protein data, but the distance matrix is generated by us.

1. Download PDB file from Internet to get the true positions of atoms.
2. Use *disk graph model* to generate distance matrix from these coordinates, with different cutoffs (6Å or 5Å) and variant noise level.

$$d_{ij} = d_{ij}(1 + noise * randn)$$

3. Make numerical experiments with these distance matrices. Distances are the only information we need to implement our algorithm.
4. Examine the RMSD error of calculated positions with the true positions (in these experiments, we know these information).

$$RMSD(X, Y) = \min_{Q,T}\{\|X - YQ - T\|_F/\sqrt{n} : Q^{\mathrm{T}}Q = I\}$$

ID	num	per	degree			RMSD	fval		time(s)			ndet
			max	min	ave				before	post	total	
1PTQ	402	5.46	38	4	21.9	7.56e-02	2.40	3.20	0.6	0.1	0.8	402
1HOE	558	4.05	38	6	22.6	6.63e-02	3.53	4.68	0.8	0.2	0.9	558
1LFB	641	3.40	40	5	21.8	2.63e-01	4.00	5.01	1.0	0.3	1.3	641
1PHT	811	3.35	48	5	27.1	1.13e-01	6.16	7.81	1.8	0.4	2.1	806
1POA	914	2.51	39	4	22.9	1.11e-01	5.87	7.54	2.0	0.4	2.4	914
1AX8	1003	2.30	39	5	23.0	2.06e-01	6.94	8.48	2.8	0.8	3.6	1003
1F39	1534	1.47	40	5	22.6	1.08e-01	9.69	12.69	7.4	0.7	8.1	1534
1RGS	2015	1.12	41	3	22.6	1.64e-01	13.16	16.78	15.3	1.2	16.6	2010
1KDH	2846	0.83	43	4	23.6	3.19e-01	72.22	24.61	43.7	4.3	48.0	2846
1BPM	3671	0.66	42	3	24.4	1.17e-01	25.83	33.27	86.9	1.8	88.7	3668
1RHJ	3740	0.65	40	4	24.4	1.01e-01	26.39	34.15	90.4	2.5	92.9	3740
1HQQ	3944	0.60	40	3	23.7	1.75e-01	26.55	34.78	109.9	2.7	112.6	3938
1TOA	4292	0.56	39	3	24.0	8.73e-02	29.94	38.59	147.1	3.2	150.3	4280
1MQQ	5681	0.44	44	5	25.2	1.73e-01	42.44	53.18	323.1	5.5	328.6	5681

Table 1 : cutoff=5Å, noise=0.01

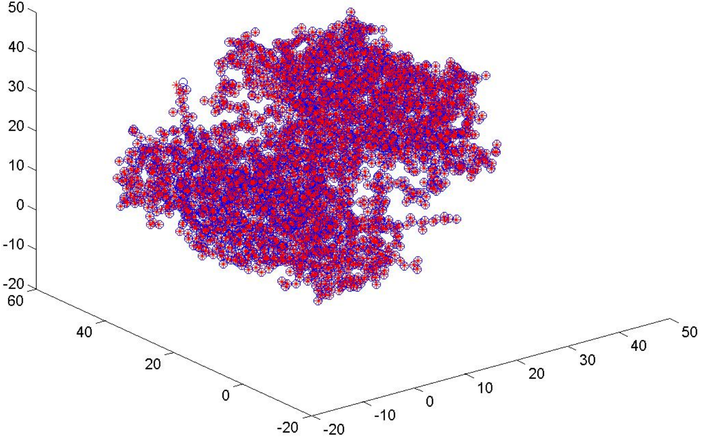
ID	num	per	degree			RMSD	fval	time(s)			ndet
			max	min	ave			before	post	total	
1PTQ	402	8.79	61	6	35.3	4.96e-02	6.00	1.2	0.1	1.3	402
1HOE	558	6.55	65	11	36.5	3.51e-02	8.90	1.8	0.2	1.9	558
1LFB	641	5.57	59	8	35.7	5.53e-02	9.70	2.0	0.2	2.2	641
1PHT	811	5.37	75	7	43.5	7.34e-02	15.82	4.1	0.3	4.3	811
1POA	914	4.07	67	8	37.2	5.22e-02	15.04	3.9	0.3	4.2	914
1AX8	1003	3.74	59	7	37.5	4.25e-02	16.29	4.6	0.3	5.0	1003
1F39	1534	2.43	62	7	37.2	4.87e-02	25.55	10.6	0.5	11.2	1534
1RGS	2015	1.87	66	4	37.7	6.28e-02	33.92	19.4	1.0	20.4	2015
1KDH	2846	1.36	64	5	38.8	6.85e-02	49.81	47.7	1.6	49.2	2846
1BPM	3671	1.12	64	4	40.9	4.24e-02	68.33	96.6	2.0	98.6	3671
1RHJ	3740	1.10	61	5	41.2	3.21e-02	70.48	103.1	2.1	105.2	3740
1HQQ	3944	1.00	64	5	39.5	5.35e-02	69.81	121.7	2.4	124.1	3944
1TOA	4292	0.94	62	4	40.1	5.80e-02	77.70	151.9	2.6	154.6	4292
1MQQ	5681	0.75	66	7	42.4	3.34e-02	110.56	334.7	3.8	338.5	5681

Table 2 : cutoff=6Å, noise=0.01

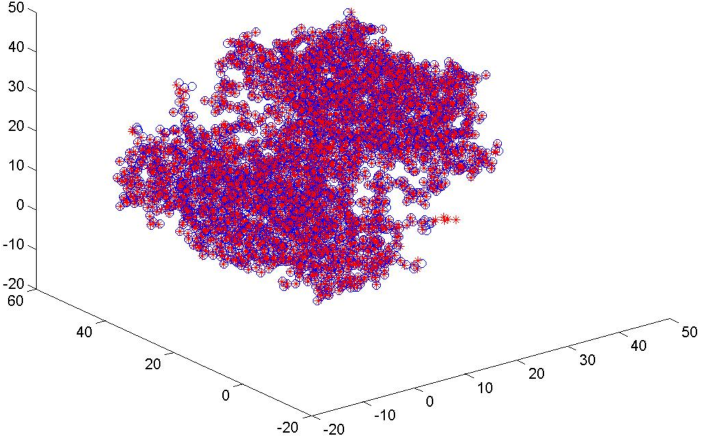
ID	num	per	degree			RMSD	fval	time(s)			ndet
			max	min	ave			before	post	total	
1PTQ	402	8.79	61	6	35.3	2.44e-01	149.64	2.4	0.3	2.7	402
1HOE	558	6.55	65	11	36.5	1.76e-01	221.95	4.0	0.3	4.3	558
1LFB	641	5.57	59	8	35.7	2.46e-01	241.74	4.6	0.5	5.1	641
1PHT	811	5.37	75	7	43.5	6.98e-01	397.14	8.0	0.6	8.6	811
1POA	914	4.07	67	8	37.2	1.77e-01	374.50	8.1	0.7	8.8	914
1AX8	1003	3.74	59	7	37.5	1.89e-01	405.16	9.3	0.7	10.0	1003
1F39	1534	2.43	62	7	37.2	2.44e-01	637.11	18.4	1.4	19.8	1534
1RGS	2015	1.87	66	4	37.7	3.13e-01	845.78	30.4	3.5	34.0	2015
1KDH	2846	1.36	64	5	38.8	2.97e-01	1240.51	69.7	4.1	73.8	2846
1BPM	3671	1.12	64	4	40.9	1.96e-01	1706.83	129.4	3.3	132.7	3671
1RHJ	3740	1.10	61	5	41.2	1.71e-01	1760.82	129.1	3.3	132.4	3740
1HQQ	3944	1.00	64	5	39.5	2.85e-01	1744.20	133.3	4.9	138.2	3944
1TOA	4292	0.94	62	4	40.1	2.36e-01	1938.62	182.6	5.0	187.6	4292
1MQQ	5681	0.75	66	7	42.4	2.04e-01	2771.67	400.3	6.1	406.4	5681

Table 3 : cutoff=6Å, noise=0.05

1HQQ: 3944



1HQQ: 3944



Outline

5 Conclusion and Future work

Conclusion and Future work

- Conclusion:
  - We proposed two new error function.
  - We designed an algorithm, and numerically proved its efficiency.
- Future work:
  - Extend the algorithm to handle distance geometry problem with distance bounds.
  - Analysis the properties of the new error functions, and design faster algorithm to

Thank you for your attention!

szl@lsec.cc.ac.cn