SOFTWARE TOOL ARTICLE

REVISED Comparing protein structures with RINspector automation in Cytoscape [version 2; referees: 3 approved]

Guillaume Brysbaert [iD], Théo Mauri, Marc F. Lensink [iD]

CNRS UMR 8576 UGSF, University of Lille, Lille, F-59000, France

Abstract

Residue interaction networks (RINs) have been shown to be relevant representations of the tertiary or quaternary structures of proteins, in particular thanks to network centrality analyses. We recently developed the RINspector 1.0.0 Cytoscape app, which couples centrality analyses with backbone flexibility predictions. This combined approach permits the identification of crucial residues for the folding or function of the protein that can constitute good targets for mutagenesis experiments. Here we present an application programming interface (API) for RINspector 1.1.0 that enables interplay between Cytoscape, RINspector and external languages, such as R or Python. This API provides easy access to batch centrality calculations and flexibility predictions, and allows for the easy comparison of results between different structures. These comparisons can lead to the identification of specific and conserved central residues, and show the impact of mutations to these and other residues on the flexibility of the proteins. We give two use cases to demonstrate the interest of these functionalities and provide the corresponding scripts: the first concerns NMR conformers, the second focuses on mutations in a structure.

Keywords

rinspector, cytoscape, protein structure, residue interaction network, centrality analysis, flexibility prediction, automation, structure ensemble

This article is included in the Cytoscape Apps gateway.

Open Peer Review

Referee Status: ✓ ✓ ✓

	Invited Referees		
	1	**2**	**3**
REVISED **version 2** published 22 Jun 2018			
version 1 published 09 May 2018	✓ report	✓ report	✓ report

1 **Sjoerd J. de Vries** [iD], Technical University of Munich, Germany

2 **Turkan Haliloglu**, Boğaziçi University, Turkey

   Ezgi Karaca, Dokuz Eylul University, Balcova, Turkey

3 **Rita Casadio**, University of Bologna, Italy

Discuss this article

Comments (0)

Corresponding author: Guillaume Brysbaert (guillaume.brysbaert@univ-lille1.fr)

Author roles: Brysbaert G: Conceptualization, Investigation, Methodology, Project Administration, Resources, Software, Supervision, Validation, Writing – Original Draft Preparation; **Mauri T**: Resources, Software; **Lensink MF**: Conceptualization, Methodology, Project Administration, Supervision, Writing – Original Draft Preparation, Writing – Review & Editing

How to cite this article: Brysbaert G, Mauri T and Lensink MF. **Comparing protein structures with RINspector automation in Cytoscape [version 2; referees: 3 approved]** *F1000Research* 2018, **7**:563 (doi: 10.12688/f1000research.14298.2)

First published: 09 May 2018, **7**:563 (doi: 10.12688/f1000research.14298.1)

Introduction

Knowledge of the structures of proteins is important to understand their function and provide a starting point for further exploration through techniques such as molecular dynamics or docking. These structures can be used directly, in the form of a PDB file, or undergo transformations prior to analysis. Residue interaction networks (RINs) are networks built from a 3D structure, where nodes represent amino acids and edges represent detected interactions between them. These networks can be generated with different tools, like RING2[1], RINerator[2] or Chimera[3], and then be imported into network visualization and analysis tools. Depending on the purpose, these network tools can be libraries available for languages like R or Python (e.g iGraph or NetworkX) or software with a graphical interface like the well-known Cytoscape[4], a reference for biological network studies. The CyREST technology[5], which is now included as a core app of Cytoscape 3.3 (and higher versions), enables further analyses, letting the user the possibility to complement analyses done in Cytoscape with scripting developments through external languages like R or Python.

We recently developed the RINspector 1.0.0[6] app for Cytoscape, which combines centrality analyses of residue interaction networks with flexibility predictions of a protein from its associated sequence through a call to the DynaMine flexibility prediction server[7,8]. Centrality analyses have been shown to identify residues important for functions, folding or allostery, as well as long range interactions (e.g. Refs 9,10). Coupled with flexibility predictions, the app enables users to highlight a subset of these central residues, which might affect the dynamics upon mutation[6].

The RINspector 1.0.0 app is convenient if one works on only a few networks, but certain centrality calculations may be CPU- and memory-intensive and require long execution times for analyses as soon as the number of networks increases. This may be the case for residue centrality analyses (RCAs) calculated on NMR data, when several conformers need to be considered. Another example is the comparison of RINs generated from a wild-type structure and several mutants. Furthermore, the app provides a score for each residue in each network and the export of a recap chart would be useful to compare between RINs. RINspector now embeds a documented application programming

interface (API) in its 1.1.0 version that provides automation of centrality calculations and flexibility predictions, thereby addressing these issues.

Here we present the automation functionalities of RINspector 1.1.0, which allows for combination of the power of Cytoscape with scripting languages such as R or Python, in order to calculate centralities, predict flexibilities, visualize the networks, and get a recap chart with scores. We present two use cases, one based on NMR data and another one based on a mutated domain structure.

Methods

Implementation

RINspector 1.1.0 is implemented in Java as an app for Cytoscape 3. It uses the JFreeChart library (Copyright 2000–2009, Object Renery Limited and Contributors) for graph representations. The DynaMine flexibility predictions are retrieved through a JSON API. Betweenness and closeness centrality values were calculated through the functions developed by Assenov *et al.*[11,12] for the Cytoscape NetworkAnalyzer core tools.

Centrality analyses and flexibility predictions can be performed via the Cytoscape graphical interface through the App/RINspector menu (see Ref. 6 for a description). They can also be called through the Cytoscape command dialog or through the CyREST command API presented here.

The RINspector 1.1.0 API relies on the REST (Representational State Transfer) architecture provided by the CyREST tools included in Cytoscape 3.3.0 (or higher), which effectively runs a web service that can only be queried on the local machine that runs Cytoscape. Communication requests follow HTTP protocol and are formatted in JSON.

Operation

The API consists of two commands, offered with documentation through the CyREST command API in Cytoscape. The commands are:

- 'centrality', which calculates centralities for each residue in the residue interaction network currently selected in Cytoscape. This command needs one parameter, which is the type of centrality calculation to perform, selectable between:

  ○ RCA (average shortest path length (ASPL) change under removal of individual nodes)

  ○ Betweenness centrality analysis – BCA

  ○ Closeness centrality analysis – CCA

RCA corresponds to the calculation proposed by 9. The ASPL of the RIN is first calculated, after which the ASPL is calculated for each network upon removal of individual single nodes. A Z-score is then computed for each node based on the change of ASPL compared to the initial one. BCA and CCA are the classical betweenness and closeness centrality calculations, both of which are followed by the calculation of a Z-score (for more details about the process, see the Supplementary materials of Ref. 6).

- 'dynamine', which queries the DynaMine server[7,8] with the sequence of the currently selected RIN. This function requires that the table of nodes in Cytoscape contains three columns: ResType, ResIndex and ResChain. The ResType column should contain the 3-letter code for each residue (e.g. ARG). The ResIndex column should contain the serial number (or index) of each residue (e.g. 153). The ResChain column should contain the chain identifier (e.g. A). These three columns are automatically created if the RIN is generated with Chimera through the structureViz2 app. One parameter has to be specified to the 'dynamine' command, namely the chain of the protein, formatted as for the ResChain column. The DynaMine server returns a S^2 flexibility score for each residue.

Each run of centrality calculation or flexibility prediction returns a score per residue. These scores appear in a dedicated column in the node table in Cytoscape and are returned as a two-columns table (node ID and score) in JSON through the REST service. The visual style of the network is also adapted (see Figure 1 and Ref. 6). The output table and/or the created columns in the node table can be further treated by a third party program written, for example, in R or Python. Table 1 presents the parameters of the POST request that are used in the provided scripts for a call of centrality or flexibility predictions and associated responses. Considering system requirements, use of a computer with at least 16 GB of memory is advised because the RCAs are memory demanding, and the bigger the structure, the more memory required. The running of use cases should not last more than a few minutes (on a Intel(R) Core(TM) i7-7820HQ CPU @ 2.90GHz, 32Go RAM, 1TB SSD, for RCA: 3m20s for use case 1, 1m47s for use case 2 ; for BCA: 10s for use case 1, 20s for use case 2; for CCA: 7s for use case 1, 17s for use case 2). Automation for RINspector requires Cytoscape 3.6.0 (or higher) and RINspector 1.1.0. The scripts we provide were developed and tested for Python 3.5.2 and R 3.2.3, with Cytoscape 3.6.1.

Use cases

We present here two examples that benefit from the API. The first one considers NMR data with 10 conformers of a yeast N-acetylglucosamine transferase, all grouped in the PDB ID 2JZC. The second one is the tetratricopeptide repeat (TPR) domain of a human O-GlcNAc transferase (OGT) (PDB ID 4GYW). Use cases files including scripts, inputs and results are available in the 'use_cases+scripts' folder of the git repository. README.txt files in each use case folder detail the steps to follow to run them.

Use case 1: 2JZC

NMR data are well suited for automation treatment because the associated PDB files usually contain multiple conformers for which RINs can be generated. We built the RINs in Cytoscape from Chimera through the structureViz2 app. Once these are
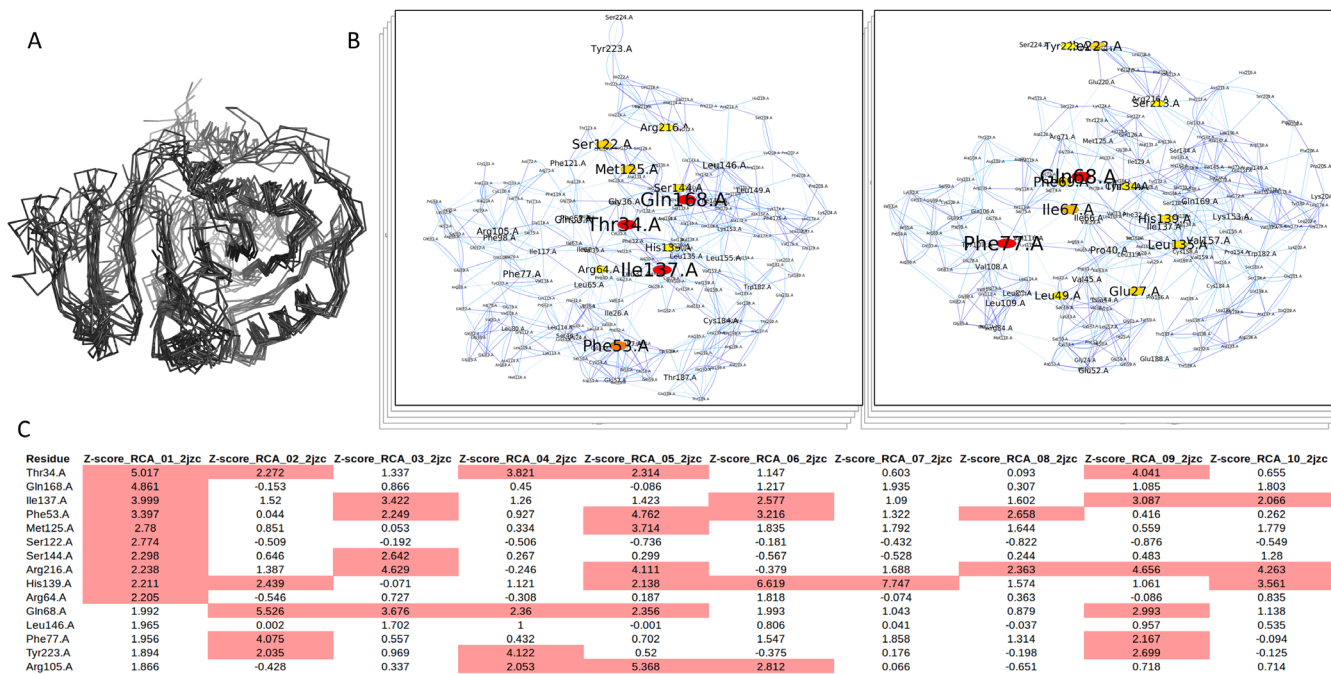

Residue	Z-score_RCA_01_2jzc	Z-score_RCA_02_2jzc	Z-score_RCA_03_2jzc	Z-score_RCA_04_2jzc	Z-score_RCA_05_2jzc	Z-score_RCA_06_2jzc	Z-score_RCA_07_2jzc	Z-score_RCA_08_2jzc	Z-score_RCA_09_2jzc	Z-score_RCA_10_2jzc
Thr34.A	5.017	2.272	1.337	3.821	2.314	1.147	0.603	0.093	4.041	0.655
Gln168.A	4.861	-0.153	0.866	0.45	-0.086	1.217	1.935	0.307	1.085	1.803
Ile137.A	3.999	1.52	3.422	1.26	1.423	2.577	1.09	1.602	3.087	2.066
Phe53.A	3.397	0.044	2.249	0.927	4.762	3.216	1.322	2.658	0.416	0.262
Met125.A	2.78	0.851	0.053	0.334	3.714	1.835	1.792	1.644	0.559	1.779
Ser122.A	2.774	-0.509	-0.192	-0.506	-0.736	-0.181	-0.432	-0.822	-0.876	-0.549
Ser144.A	2.298	0.646	2.642	0.267	0.299	-0.567	-0.528	0.244	0.483	1.28
Arg216.A	2.238	1.387	4.629	-0.246	4.111	-0.379	1.688	2.363	4.656	4.263
His139.A	2.211	2.439	-0.071	1.121	2.138	6.619	7.747	1.574	1.061	3.561
Arg64.A	2.205	-0.546	0.727	-0.308	0.187	1.818	-0.074	0.363	-0.086	0.835
Gln68.A	1.992	5.526	3.676	2.36	2.356	1.993	1.043	0.879	2.993	1.138
Leu146.A	1.965	0.002	1.702	1	-0.001	0.806	0.041	-0.037	0.957	0.535
Phe77.A	1.956	4.075	0.557	0.432	0.702	1.547	1.858	1.314	2.167	-0.094
Tyr223.A	1.894	2.035	0.969	4.122	0.52	-0.375	0.176	-0.198	2.699	-0.125
Arg105.A	1.866	-0.428	0.337	2.053	5.368	2.812	0.066	-0.651	0.718	0.714

Figure 1. Residue centrality analyses (RCAs) of NMR conformers. (**A**) Overlapped structure of the 10 conformers used to generate the residue interaction networks (RINs) for ALG13 (PDB ID 2JZC) in ribbon representation. (**B**) The RINs on which RCAs were performed; central residues are colored from yellow (Z-score = 2) to red (Z-score ≥ 4), labels and node sizes are adapted as functions of the Z-scores: the higher, the bigger; only the first two RINs of the ten calculated are shown. (**C**) Final result table, which contains the centralities of all the networks, sorted by descending order of Z-scores of the first conformer; Z-scores considered as relevant (Z-score ≥ 2) have a pink background; only the first 15 residues are displayed.

Table 1. Parameters and responses for centrality and dynamine POST requests. For each request to the REST service, the url, body and encoding format are specified. The response in both cases is a table that contains the Z-score (centrality) or the S^2 score (dynamine) associated to each node ID, in JSON format.

Centrality	dynamine
url: http://localhost:1234/v1/commands/rinspector/centrality body: one of the three methods (RCA/BCA/CCA) encoding: JSON	url: http://localhost:1234/v1/commands/rinspector/dynamine body: chain ID encoding: JSON

response:
{
  "data": {
    "308053": -0.10838826158689033,
    "308054": -0.8808526864488292,
(…)
    "308253": 0.9250226566444141
  },
  "errors": []
}
	response:
	{
	  "data": {
	    "308053": 0.737,
	    "308054": 0.73,
	(…)
	    "308253": 0.75
	  },
	  "errors": []
	}

created, the RINspector API can be queried to calculate centralities in batch. In this example, we calculate residue centrality scores for each residue and compare them between the conformations. We wrote a script in R and an equivalent one in Python that, starting from a Cytoscape session containing one RIN for each conformer, perform RCA on each RIN. The scripts then gather the Z-scores for each residue in each RIN in a single recap table, allowing for easy comparison. The app also creates a style for each RIN that permits a visual comparison (Figure 1).

Use case 2: 4GYW
Here we compare the centralities of residues when RINs are generated from structures which contain point mutations. We also compare the impact of mutations on flexibility.

We created six mutants in the tetratrico peptide repeat (TPR) domain of the O-GlcNAc transferase (OGT) by editing the PDB file of the wild type. Five mutants were asparagines to alanines (N322A, N325A, N356A, N390A, N424A), the sixth contained all five point mutations. The OGT is an enzyme that catalyzes the transfer of a single N-acetylglucosamine from UDP-GlcNAc to a serine or threonine amino acid (called O-GlcNAcylation). These five asparagines have been shown to decrease the efficiency of the OGT enzymatic activity on a category of peptides when simultaneously mutated into alanines[13]. We generated the RINs (without ligands) for the wild-type structure and each of the six variants. We wrote a script that, starting from a Cytoscape session containing these RINs, calculates residue centralities and predicts flexibilities for each structure. As in the use case 1, the results of centralities are gathered in a common table. In addition, the S^2 flexibility scores are also gathered in a recap table and in a single plot to permit easy comparison (Figure 2). These tables and plot allow for comparison between the centralities, to see which of the mutations have an impact on which centralities compared to the wild type, and to see the impact of each mutation on the backbone flexibility of the TPR domain.

Discussion
Comparisons between structures, and more precisely between residues in structures, are useful for identification of one or a subset of amino acids that are crucial for the folding or functions of proteins. The automation functionalities of RINspector 1.1.0 provided through CyREST in Cytoscape make these comparisons easier. Indeed, they permit to perform centrality analyses and flexibility predictions of residue interaction networks generated from structures of proteins, in batch mode from an external script. The results can then be exploited using the libraries available for the language of the script (usually R or Python).

In our cases, we treated at most ten networks at a time, which is reasonable. This number can of course be higher, but depending on the size of the structure/network, it may be difficult to deal with a substantially higher number of networks. Indeed, while BCAs and CCAs are usually performed within a few seconds, RCAs are memory demanding and running many of them may quickly fill the RAM of the system. In such cases we recommend to perform all the calculations through a workflow outside Cytoscape. The CyREST core or command API can nevertheless still be called in the workflow, e.g. for the generation of networks visualizations. RCAs and BCAs usually give many central residues which show various degrees of overlap. Depending on the objective of the analyses, it may be more relevant to consider the results of one centrality, the intersection, or the union of these selections. However, CCAs generally result in few residues with Z-scores ≥ 2. If the user is interested in this specific measurement, we advise to visualize directly the closeness centrality values and not the Z-scores (with NetworkAnalyzer tools in Cytoscape or RINalyzer app[2] for instance).

The RINspector 1.1.0 API can be used in conjunction with other apps, especially with the structureViz2 app, which connects Chimera to Cytoscape, because it permits to open a structure in Chimera and work with it. This connection allows for the easy
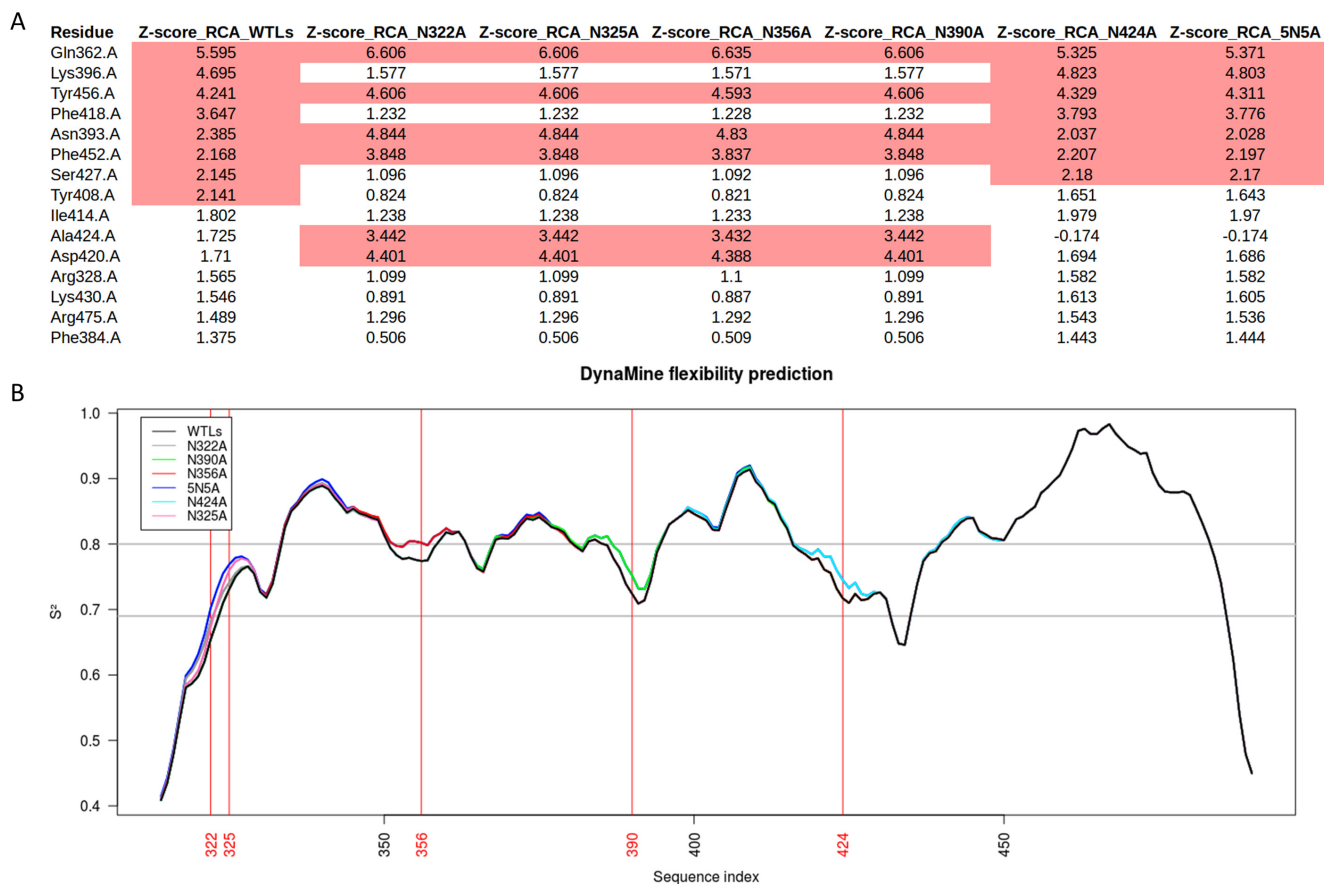
A

Residue	Z-score_RCA_WTLs	Z-score_RCA_N322A	Z-score_RCA_N325A	Z-score_RCA_N356A	Z-score_RCA_N390A	Z-score_RCA_N424A	Z-score_RCA_5N5A
Gln362.A	5.595	6.606	6.606	6.635	6.606	5.325	5.371
Lys396.A	4.695	1.577	1.577	1.571	1.577	4.823	4.803
Tyr456.A	4.241	4.606	4.606	4.593	4.606	4.329	4.311
Phe418.A	3.647	1.232	1.232	1.228	1.232	3.793	3.776
Asn393.A	2.385	4.844	4.844	4.83	4.844	2.037	2.028
Phe452.A	2.168	3.848	3.848	3.837	3.848	2.207	2.197
Ser427.A	2.145	1.096	1.096	1.092	1.096	2.18	2.17
Tyr408.A	2.141	0.824	0.824	0.821	0.824	1.651	1.643
Ile414.A	1.802	1.238	1.238	1.233	1.238	1.979	1.97
Ala424.A	1.725	3.442	3.442	3.432	3.442	-0.174	-0.174
Asp420.A	1.71	4.401	4.401	4.388	4.401	1.694	1.686
Arg328.A	1.565	1.099	1.099	1.1	1.099	1.582	1.582
Lys430.A	1.546	0.891	0.891	0.887	0.891	1.613	1.605
Arg475.A	1.489	1.296	1.296	1.292	1.296	1.543	1.536
Phe384.A	1.375	0.506	0.506	0.509	0.506	1.443	1.444

B

Figure 2. Residue Centrality Analyses (RCAs) of the TPR domain of the human O-GlcNAc transferase (PDB ID 4GYW). (**A**) Final result table showing the wild type TPR domain of the human OGT and the 6 mutants N322A, N325A, N342A, N356A, N390A and 5N5A (containing all five mutations), sorted by descending order of Z-scores of the wild type; Z-scores considered as relevant (Z-score ≥ 2) have a pink background; only the first 15 residues are shown. (**B**) DynaMine flexibility graphs of the wild type (WTLs) and each of the mutants; X=sequence index, Y=S^2, the higher S^2 value, the more rigid the backbone is predicted to be; grey horizontal straight lines delimit a context dependent zone; mutated residues are highlighted with red vertical straight lines.

generation of RINs and the possibility of synchronization of colors between network and structure, which is a welcome feature for the generation of images. In this respect we would also like to point out the RINalyzer app, which enables the user to layout the RIN in function of coordinates of residues displayed in Chimera.

In the use case 2, DynaMine results were exploited to build a chart for comparison of flexibilities between structures. For such analyses the proposed script is particularly interesting, as computation resources are not a limiting factor. In most cases, however, only a few mutations are really of interest and the result panel designed for DynaMine in RINspector may be more convenient to visualize the effect of these mutations directly and select one or several for protein design. The user should also be aware that the sequence sent to the DynaMine server is built from the RIN/structure, which means that missing residues will simply be skipped in the DynaMine flexibility graph.

Other use case examples might be the effect of one or several ligands in a structure by generating RINs with or without these

ligands, the comparison of centralities of different structures of the same complexes or the comparison of the flexibility of structures of several orthologs. Our scripts constitute starting points to perform such analyses with recap tables and charts as output.

Future developments
We plan to extend the interaction between other languages and Cytoscape complementing batch centrality analyses and flexibility predictions with automatic generation of RINs and integration of conservation data.

Summary
The RINspector 1.1.0 API permits scripting of centrality analyses and flexibility predictions. The automation provided bridges between Cytoscape and other languages, such as R or Python, to prepare data, run batch analyses and treat output data. It enlarges the possibilities of treatments that were initially given by the app in particular to compare centralities and flexibilities between multiple structures. We provide R and Python scripts that illustrate two use cases and that are to be seen as starting scripts for more elaborate analyses.

Data and software availability

1. RINspector is available from the Cytoscape App Store: http://apps.cytoscape.org/apps/rinspector

2. Latest source code and example files are available at: https://sourcesup.renater.fr/scm/?group_id=3888

3. Archived source code as at time of publication: https://doi.org/10.5281/zenodo.1292457[14].

4. Software license: CeCILL; version 2.1: http://www.cecill.info

References

1. Piovesan D, Minervini G, Tosatto SC: **The RING 2.0 web server for high quality residue interaction networks.** *Nucleic Acids Res.* 2016; **44**(W1): W367–74.
   PubMed Abstract | **Publisher Full Text** | **Free Full Text**

2. Doncheva NT, Klein K, Domingues FS, *et al.*: **Analyzing and visualizing residue networks of protein structures.** *Trends Biochem Sci.* 2011; **36**(4): 179–82.
   PubMed Abstract | **Publisher Full Text**

3. Pettersen EF, Goddard TD, Huang CC, *et al.*: **UCSF Chimera--a visualization system for exploratory research and analysis.** *J Comput Chem.* 2004; **25**(13): 1605–12.
   PubMed Abstract | **Publisher Full Text**

4. Shannon P, Markiel A, Ozier O, *et al.*: **Cytoscape: a software environment for integrated models of biomolecular interaction networks.** *Genome Res.* 2003; **13**(11): 2498–504.
   PubMed Abstract | **Publisher Full Text** | **Free Full Text**

5. Ono K, Muetze T, Kolishovski G, *et al.*: **CyREST: Turbocharging Cytoscape Access for External Tools via a RESTful API [version 1; referees: 2 approved].** *F1000Res.* 2015; **4**: 478, [cited 2018 Mar 15].
   PubMed Abstract | **Publisher Full Text** | **Free Full Text**

6. Brysbaert G, Lorgouilloux K, Vranken W, *et al.*: **RINspector: a Cytoscape app for centrality analyses and DynaMine flexibility prediction.** *Bioinformatics.* 2018; **34**(2): 294–296.
   PubMed Abstract | **Publisher Full Text** | **Free Full Text**

7. Cilia E, Pancsa R, Tompa P, *et al.*: **From protein sequence to dynamics and disorder with DynaMine.** *Nat Commun.* 2013; **4**: 2741.
   PubMed Abstract | **Publisher Full Text**

8. Cilia E, Pancsa R, Tompa P, *et al.*: **The DynaMine webserver: predicting protein dynamics from sequence.** *Nucleic Acids Res.* 2014; **42**(Web Server issue): W264–70.
   PubMed Abstract | **Publisher Full Text** | **Free Full Text**

9. del Sol A, Fujihashi H, Amoros D, *et al.*: **Residues crucial for maintaining short paths in network communication mediate signaling in proteins.** *Mol Syst Biol.* 2006; **2**: 2006.0019.
   PubMed Abstract | **Publisher Full Text** | **Free Full Text**

10. Hu G, Yan W, Zhou J, *et al.*: **Residue interaction network analysis of Dronpa and a DNA clamp.** *J Theor Biol.* 2014; **348**: 55–64.
    PubMed Abstract | **Publisher Full Text**

11. Assenov Y, Ramírez F, Schelhorn SE, *et al.*: **Computing topological parameters of biological networks.** *Bioinformatics.* 2008; **24**(2): 282–4.
    PubMed Abstract | **Publisher Full Text**

12. Doncheva NT, Assenov Y, Domingues FS, *et al.*: **Topological analysis and interactive visualization of biological networks and protein structures.** *Nat Protoc.* 2012; **7**(4): 670–85.
    PubMed Abstract | **Publisher Full Text**

13. Rafie K, Raimi O, Ferenbach AT, *et al.*: **Recognition of a glycosylation substrate by the O-GlcNAc transferase TPR repeats.** *Open Biol.* 2017; **7**(6): pii: 170078.
    PubMed Abstract | **Publisher Full Text** | **Free Full Text**

14. Brysbaert G, Mauri T, Lensink MF: **RINspector source code (Version 1.1.0).** *Zenodo.* 2018.
    Data Source

Open Peer Review

Current Referee Status: ✔ ✔ ✔

Version 1

Referee Report 18 June 2018

✔ Rita Casadio

Biocomputing Group, Department of Pharmacy and Biotechnology, University of Bologna, Bologna, Italy

Comparing protein structure with RINspector is an intersting application that allows an easy determination of crucial residues in protein structures. It can be useful for pinpointing specific positions in the protein backbone for further experimental investigations.

The paper clearly explains the procedure, and clarifies with test cases the relevance of the application. Software is detailed and reproducible. Figures are clear and appropriate.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: No competing interests were disclosed.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Author Response 20 Jun 2018
Guillaume Brysbaert, CNRS, France

Dear Dr Casadio,

Thank you for the review and your interest in our work. We are happy to see that you accept the manuscript.

Best regards

Competing Interests: No competing interests were disclosed.

Referee Report 12 June 2018

doi:10.5256/f1000research.15559.r33872

Turkan Haliloglu [1], **Ezgi Karaca** [2]

[1] Polymer Research Center & Department of Chemical Engineering, Boğaziçi University, Istanbul, Turkey
[2] International Biomedicine and Genome Institute, Dokuz Eylul University, Balcova, Izmir, Turkey

The RINspector app allows network and flexibility analysis of protein structures, which plausibly has the capacity to disclose functionally critical residues. The paper by Brysbaert et al. explains the updated (automated) version of the RINspector app. The new version is stated to work more efficiently than previous one, especially while treating multi-conformers. The paper could further be improved, if the following points are addressed.

- The differences between the previous and current version of the program are not clearly stated. It is confusing what is new and what is updated, especially for someone who is not familiar with the RINspector app. Maybe the authors could mention the version numbers explicitly when they make a comparison.
- There is a tutorial provided for the initial version of RINspector. A similar tutorial should also be prepared for the use cases presented in this paper. In this way, the users can exactly reproduce what is stated in the manuscript.
- Relevant links to the software, the use case folders and scripts should be cited within the text. In the current version, it is hard to link the source code files with the options/conditions stated in the paper.

Overall, RINspector app stands a useful resource that would corroborate structural and functional studies in the community.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: No competing interests were disclosed.

Referee Expertise: Modeling and simulation of macromolecules. Protein dynamics and allostery.

We have read this submission. We believe that we have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Author Response 20 Jun 2018

Guillaume Brysbaert, CNRS, France

Dear Dr Haliloglu and Dr Karaca,

Thank you for your interest in our article and your constructive comments. We address them point by point.

- The differences between the previous and current version of the program are not clearly stated. It is confusing what is new and what is updated, especially for someone who is not familiar with the RINspector app. Maybe the authors could mention the version numbers explicitly when they make a comparison.

→ The versions of RINspector were specified in the text to avoid confusion.

- There is a tutorial provided for the initial version of RINspector. A similar tutorial should also be prepared for the use cases presented in this paper. In this way, the users can exactly reproduce what is stated in the manuscript.

→ We included a 'README.txt' file in each use case folder to explain how to run the example scripts. The point of the paper being the automation of RINspector, we don't explain how to generate the RINs but we provide a python script called 'creation_RIN_NMR.py' in the use case 1 that permits to generate the scripts with structureViz2/Chimera.

- Relevant links to the software, the use case folders and scripts should be cited within the text. In the current version, it is hard to link the source code files with the options/conditions stated in the paper.

→ In the 'Use cases' part, we added the information that the use cases files (inputs/scripts/outputs) are available in the git repository and added a hyperlink to this folder in the git.

Best regards

Referee Report 12 June 2018

✔

Sjoerd J. de Vries 🆔

MTi, UMRS 973, Physics Department T38, Technical University of Munich, Garching, Germany

Brysbaert et al. describe an API for their RINspector tool for the analysis of residue interaction networks (RINs). This API allows RIN analysis in an automated fashion (batch mode), controlled from any programming language that can perform HTTP requests. The authors provide two use cases as an example.

The article is sound and well-written. I have only minor comments pertaining the didactic value of the paper: I feel the authors could lower the barrier for users of the software, by explaining better what it is good for and how to get it running.

- The running time of the use cases should be reported, to demonstrate the amount of human time saved.
- The use case example scripts are available from the Git repository, but this is not mentioned, and the repository itself is not easy to browse. Direct links would be better.
- The software relies heavily on Cytoscape, its app store, and it's cyREST technology, and the authors assume that the reader has good knowledge of these. However, Cytoscape is not much used in the protein structure community. I recommend that the authors include a link to a short install manual for Cytoscape, cyREST and RINspector, plus any other necessary steps that must be performed before the example scripts can be run.
- It deserves some explanation that the RINspector API is not meant to (and in fact, unable to) provide any kind of web service. The reason why might be well-known within the Cytoscape community, but again, Cytoscape is not very well known among structural biologists, and I have never used it myself. From what I understand, the cyREST technology is very poorly named, since it is not a proper (i.e. stateless) REST protocol at all. Instead, it seems to be a technology to send state-manipulating commands (load data; analyze current data; etc.) over HTTP to a running instance of Cytoscape. The RINspector API follows this state-manipulating paradigm. In addition, it seems that Cytoscape cannot run behind a web server because it always runs a GUI.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

Competing Interests: No competing interests were disclosed.

Referee Expertise: computational structural biology, structural bioinformatics

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Author Response 20 Jun 2018

Guillaume Brysbaert, CNRS, France

Dear Dr de Vries

Thank you for your interest of our manuscript and your helpful remarks. We took them in consideration and updated the manuscript. Here are our replies to your comments:
- The running time of the use cases should be reported, to demonstrate the amount of human time saved.

→ Running times were added in the text in the 'Operation' part.
- The use case example scripts are available from the Git repository, but this is not mentioned, and the repository itself is not easy to browse. Direct links would be better.

→ We added the information that the example files are available in the git repository. We specified the relative folder on the git repository and set a link to the folder in tree visualisation.
- The software relies heavily on Cytoscape, its app store, and it's cyREST technology, and the authors assume that the reader has good knowledge of these. However, Cytoscape is not much used in the protein structure community. I recommend that the authors include a link to a short install manual for Cytoscape, cyREST and RINspector, plus any other necessary steps that must be performed before the example scripts can be run.

→ In each use case folder, we included a README.txt file that details the required steps to run the scripts. We added this information in the 'Use cases' part of the article. CyREST is included in the last versions of Cytoscape, that's why the README.txt files don't contain any mention about CyREST installation.
- It deserves some explanation that the RINspector API is not meant to (and in fact, unable to) provide any kind of web service. The reason why might be well-known within the Cytoscape community, but again, Cytoscape is not very well known among structural biologists, and I have never used it myself. From what I understand, the cyREST technology is very poorly named, since it is not a proper (i.e. stateless) REST protocol at all. Instead, it seems to be a technology to send state-manipulating commands (load data; analyze current data; etc.) over HTTP to a running instance of Cytoscape. The RINspector API follows this state-manipulating paradigm. In addition, it seems that Cytoscape cannot run behind a web server because it always runs a GUI.

→ RINspector is an app for the standalone Cytoscape software that uses the REST service provided by the CyREST tool inside Cytoscape. As such, contrarily to the usual REST services designed for a web usage, Cytoscape is a standalone tool with a GUI that has to be run on a single computer to start the REST service that is only accessible on this computer. Requests are done thanks to the HTTP protocol and are formatted in JSON. We added explanations in the

'Implementation' part. The REST architecture that is used is stateless in the sense that each request is independent from another and no state is conserved by the local server.

Best regards

Competing Interests: No competing interests were disclosed.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias

- You can publish traditional articles, null/negative results, case reports, data notes and more

- The peer review process is transparent and collaborative

- Your article is indexed in PubMed after passing peer review

- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com