D3.1: Software for generators and performance report

Project acronym: *DYNANETS*

Project full title: Computing Real-World Phenomena with Dynamically Changing Complex Networks.

Grant agreement no.: 233847

Due-Date:	M12
Delivery:	M14 resubmission M17
Lead Partner:	ISI
Dissemination Level:	Public
Status:	Re-submitted
Approved:	Q Board, Project Steering Group
Version:	V2.2 re-submission Oct 2010

DOCUMENT INFO

0.1 Authors

Date and version number	Author	Details
21-10-2010	Lorenzo Isella Vittoria Colizza George Kampis Laszlo Gulyas Arnaud Grignard Rick Quax	Revision of D3.1 that was submitted to EC but was rejected.
28/10/2010 v2.2	George Kampis	Editing, Revisions QAB

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

1 EXECUTIVE SUMMARY

Workpackage 3 (WP3) “Complex Network Modeling and Computational Representations” aims at developing software tools for the structural and temporal modeling of dynamical networks. Within the scope of WP3, the current Deliverable 3.1 (D3.1) provides the computational tools for the structural analysis and generation of dynamical networks. Some of these tools are meant to be generic, in order to provide a theoretical framework for the fundamental understanding of dynamical networks across disciplines, whereas others are necessarily domain-specific, aiming at the understanding of specific mechanisms that are present in given settings/phenomena (e.g. the dynamics of conversational encounters and interactions, or the dynamics of HIV spreading on sexual contacts). The research carried out during the first twelve months of WP3 lead to the development of the following new tools by the DynaNets consortium:

- tools for the representation and statistical analysis of dynamical networks, contributed by ISI; given that dynamical networks may be generated from a set of statistical parameters or from collected data, this tool was applied to the output of network generating models (see following points) and to a specific application of data-driven networks of face-to-face human interactions, from high spatial and time resolution data collected in different social environments [1, 2];
- elementary dynamics network (EDNs) models, contributed by ColBud and leading to the publication in Ref. [3], which can be seen as the dynamical counterparts of classical network models (Barabasi-Albert, Erdos-Renyi etc...) where the topology is continuously evolving in time through various mechanisms and the aim is the investigation of the resilience properties of dynamically evolving networks;
- a complex network simulator, contributed by UvA, to efficiently model a dynamical process occurring on dynamical networks where nodes and links are characterized by features that may evolve in time; given the intensive computational complexity induced by the coupled dynamics, the tool is developed to best optimize its computational efficiency; the application is the study of an HIV epidemic circulating on the sexual network of men having sex with men, based on data collected within two cohort studies, conducted in Amsterdam and San Francisco [4, 5];
- a multiscale framework for the study of dynamical processes occurring on networks that unfold at multiple scales; this tool was developed within the CoSMo computational toolkit, contributed by ENS, to model the spread of infectious

diseases at the individual and population level, taking into account spatially structured populations and mobility processes;.

- a modeling framework to generate dynamical and bursty contact networks, contributed by ISI and leading to the publication in Ref. [6], consisting of agents in social interaction and capable of reproducing the salient features observed in the aforementioned empirical dataset of face-to-face interactions.

2 Contributors

- ISI: workpackage leader. ISI contributed newly developed software tools for the statistical analysis of dynamical networks and an agent based model for contact networks, both complemented with tools for network static visualization. See sections 3.1. and 3.5.
- UvA: partner. UvA contributed a new multiscale software simulator, SEECN, where the nodes and edges have specified properties which dictate the dynamics of the network over time. See section 3.3.
- ColBud: partner. ColBud contributed new EDNs models for investigating the robustness of dynamical networks. See section 3.2.
- ENS: partner. ENS integrated several new network models (random network, small world network, etc...) and disease spreading models in their legacy CoSMo toolkit. See section 3.4.

3 Software tools

WP3 revolves around designing and implementing a computational framework for modeling evolving networks. The aim is to provide flexible tools of analysis for the generation and representation of dynamical networks.

In the following, we will describe in detail the chosen representation of dynamical networks and the tools provided by WP3 complemented by several applications, but we stress that the range of applicability of the computational tools is by no means limited to the case studies presented in this deliverable.

3.1 Representation of dynamical networks (contribution by ISI)

The efficient representation of an evolving network is crucial for the performance of the tools for data analysis. The challenge of representing dynamic networks is a newly

developing problem with few ready made solutions available. Due to its simplicity and efficiency, we resorted to a time-dependent edge list. Every node is identified by an integer number chosen as its ID. The interaction between nodes is then mapped into network edges. The interpretation of the interaction establishing a link between two nodes depends of course on the system in examination (e.g. a face-to-face interaction for the human interaction networks and a sexual contact when dealing with the HIV infection network). The evolving network is then expressed as a time-dependent edge list where at every (discrete) time $t_i = i\delta$ we report the ID of the interacting nodes. Here δ stands for a time window corresponding to the minimum time scale one can resolve in the system. Of course this time scale depends on the system under scrutiny (for instance, it would correspond to 20s for the networks of human contacts and to 3 months for the HIV spreading simulations). An isolated node at time t_i is expressed as a self-interacting node like for example node 1300 at time t_0 in Table 1.

The time-dependent edge list representation of a dynamic network amounts to stitching together several consecutive network snapshots. One can easily track e.g. a single node (for instance to investigate its contact duration distribution, the number of different nodes he established a contact with and so on) by simply looking for the occurrences of its corresponding ID number. This representation used in our innovative approach has proved particularly useful to generate aggregated networks of human contacts (discussed in detail in the following), but it is a quite general representation of an evolving network.

Table 1: Representation of a dynamic network as a time-dependent edge list.

Time	ID of Node A	ID of Node B
t_0	1100	1200
t_0	1300	1300
...	...	...
t_1	900	1000
t_1	1100	1500
t_1	1100	1200
…	…	…
t_i	1400	1800
…	…	...

The time-dependent edge list Table 1 allows one easily obtain the aggregated network along a given time interval. Indeed aggregating the network in the time interval $[t_{ini}, t_{fin}]$

amounts simply to selecting the corresponding time window in the time-dependent edge list.

In the aggregated network an edge is drawn between node A and node B if at least one contact was detected between those nodes during the aggregation time interval. The duration of an interaction between node A and node B can then be easily calculated by simply counting the occurrences of A ↔ B edges multiplied by the time window δ. We stress that the strategy described above is by no means limited to the datasets analyzed in this deliverable, but it can be applied to any dynamic network whose time-dependent edge list is known and for which it makes sense to introduce the concept of link duration. Furthermore, finding the neighbours of node A (and therefore the degree distribution of the aggregated network when we iterate the procedure on all the nodes) simply amounts to identifying the number of unique A ↔ B links, with B = A, in the aggregated network. Finally, the calculation of the life-time of a node in the aggregated network is reduced to identifying the first and last occurrence of that node in the aggregated network.

Furthermore the time-dependent edge list representation allows one to split different tasks. For instance, network rewiring requires necessarily the previous generation and storage in memory of the corresponding aggregated network. On the other hand, as pointed out before, the calculation of several other quantities of interest like e.g. the contact duration distribution essentially amounts to a series of array manipulations on the time-dependent edge list. As a consequence, we are able to separate to some extent the analysis of network topology from the analysis of some network temporal properties, thus saving computational time since both tasks can be carried out independently.

In order to validate and stress test the tools for statistical analysis, we focused on the data about human interactions collected in two strongly different contexts. The first one is an exhibition held at the Science Gallery in Dublin [1], Ireland, from April 17th to July 17th, 2009 (hereafter referred to as SG). The second is a scientific conference (Hypertext 2009, or HT09 [2]), hosted by the Institute for Scientific Interchange Foundation in Turin, Italy, from June 29th to July 1st , 2009.

Analysis of large datasets about human activities and interactions have long been limited by the difficulty of gathering information. The ever-increasing use and availability of digital information are however widely enabling the collection and analysis of massive amounts of data about many aspects of human behaviour. In particular, modern portable sensing devices allow researchers to collect and data mine massive amounts of information about human activities, thus changing substantially the approach to the study

of human and social interaction. Bluetooth or wifi technology give access to proximity patterns [7,8,9,10,11], and even face-to-face interactions can be resolved [12,13,14,15] both spatially and temporally. In this context, the use of a description in terms of complex networks has been a widely successful tool [16,17,18,19], thanks to its versatility.

Intuitively, interactions among conference participants differ from interactions among museum visitors, and the concerned individuals have very different goals in both settings. The deployed infrastructure to collect data about human interaction uses radio frequency devices (RFID) embedded into conference badges engaging in ultra-low power bidirectional packet exchange, as described in [12,13,14,15]. Conference participants at HT09 and museum visitors volunteer to carry these wearable RFID devices. The deployments at the Science Gallery in Dublin and at HT09 conference in Turin involved a vastly different number of individuals and stretched along different time scales. The former lasted about three months and recorded the interactions of more than 14,000 visitors (more than 230,000 recorded face-to-face contacts), whereas the latter took place during three days involving about 100 conference participants (about 10,000 contacts). The software tools we contributed enable the analysis of the collected data and the exploration of the network structural and dynamical properties (degree distribution, assortativity, weight and strength distribution to name but a few.

We will hereafter refer to interaction among tags A and B as a shorthand to mean interaction among two distinct individuals carrying tags A and B. For the aggregated networks of human contacts, each edge is naturally weighted by the total duration of the various contact events occurred between these tags, i.e., by the total time during which the individuals have been in contact in the chosen aggregation time window. The choice of daily time windows seems quite natural as it would represent for instance a typical time scale for a representation of social networks based on surveys of the participants, in which each participant would (ideally) record who s/he has encountered during the day. Such a choice of the time window, albeit natural, is by no means compulsory. For instance, we have also studied the museum data along longer periods (weeks, months) to investigate the stationarity of some properties of the collected data. Shorter aggregation times of the order of a few minutes are also useful for instance in order to investigate the variation of the number of museum visitors within a single day.

Examples of aggregated networks from both a day at HT09 and a few representative days at the SG are given in Fig. 1, where we highlight the network diameter. The

aggregated network at the HT09 exhibits a short diameter and a high connectivity, whereas the aggregated networks at the SG have a much longer diameter and are often split into two connected components (CC). After aggregating the network on the desired time interval, the tools developed for WP3 can be used to perform a number of statistical analyses on the network. As an example, we calculate the degree distribution P(k), i.e. the probability that a randomly chosen node has k neighbours, which is one of the most important quantities to characterize a network topology.

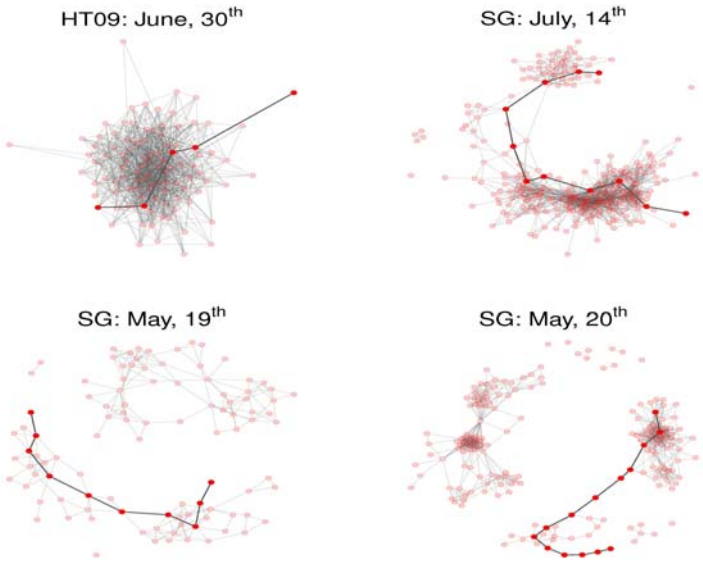

Figure 1: Aggregated network along one day (with highlighted network diameter). Clockwise from top: HT09 conference and three representative days at the Science Gallery Museum in Dublin.

In Fig. 2 we show the degree distributions obtained by gathering together the data from the aggregated networks on a 24-hour basis for the whole duration of the deployments in the museum (right) and at the HT09 conference (left). For the museum data, we left out the (few) isolated nodes which obviously contribute only with k = 0 to the degree distribution.

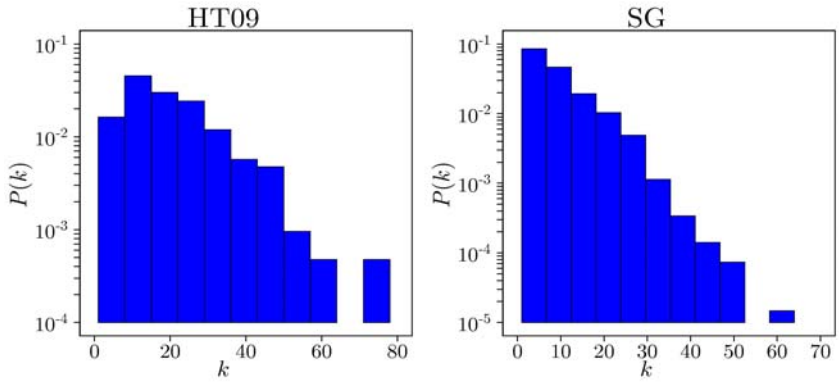

Figure 2: Left: degree distribution P (k) at HT09. Right: as in left for the aggregated networks from the SG.

The tools for structural network analysis include the generation a randomized version of the original network by resorting to the rewiring procedure described in Ref. [20], which preserves the degree distribution P (k) while destroying correlations. A randomized network is often needed as a null model to carry out statistical analysis. In Fig. 3 we plot a single realization of the null model for the networks in the top row of Fig. 1.

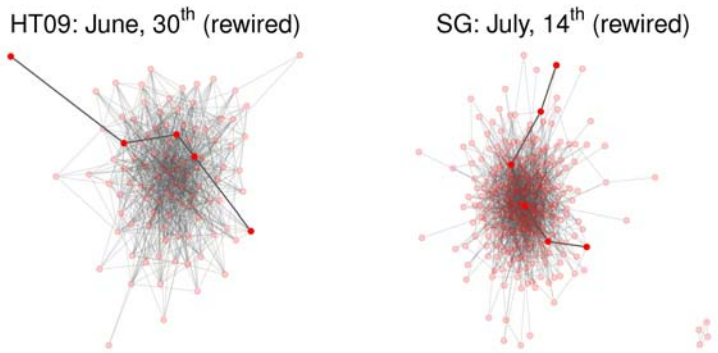

Figure 3: Randomized version of the aggregated networks in the top row of Fig. 1. Left: HT09, June, 30th . Right: SG, July, 14th .

One notices that the rewired version of the aggregated network at the HT09 is very similar to the original aggregated network, whereas the null model for the aggregated network in the museum on July, 14th is more compact than the original network and exhibits a much shorter diameter. Similar considerations hold for the other aggregated networks of the museum setting.

The tools developed for WP3 allow one to represent the network longitudinal dimension by studying several temporal properties of the nodes and/or the links. For instance, in Fig. 4 (top) we show the distribution of visit durations obtained collecting the data for the whole duration of the experiment at the SG and a fit to a lognormal distribution (red line) with geometric mean around μ=35 minutes. This shows that, unlike the case of the conference, here one can meaningfully introduce the concept of a characteristic visit duration, which turns out to be well below the cut-off imposed by museum opening hours.

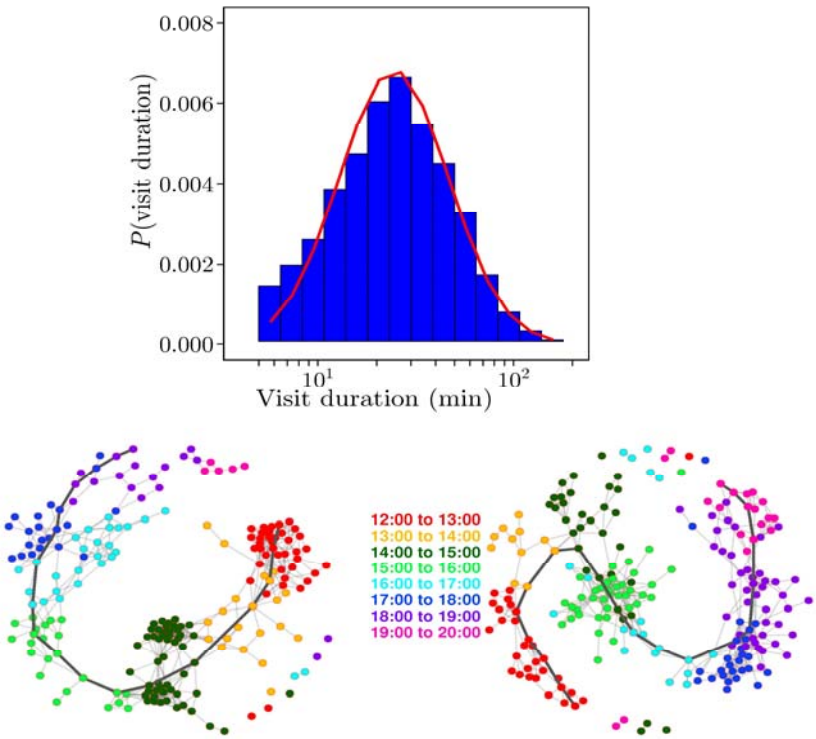

Figure 4: Top: visit duration distribution at the SG, Dublin, and fit to a lognormal distribution (red line). Bottom: for two aggregated networks at the SG, every node is colored according to the individual's entry time slot. The network diameter is highlighted.

The existence of characteristic visit duration sheds light on the elongated aspect of the aggregated networks of visitor interactions (see Fig. 1). Indeed museum visitors are unlikely to interact directly with other visitors entering the museum more than an hour after them, thus preventing the aggregated network from exhibiting small-world properties. In Fig 4 (bottom) we show the aggregated networks for two different days at

the SG. Each node is coloured according to the individual's entry time slot. Again, the network diameter is highlighted. The network diameter connects early visitors with late visitors, thus exhibiting a temporal beside a topological meaning. This also shows that network topology and network longitudinal dimension are deeply interwoven. Many other temporal quantities of interest can be investigated to shed light on the human dynamics. For instance, we can calculate the contact

duration and cumulative contact duration (weight w_{ij}) distribution, as shown in Fig. 5.

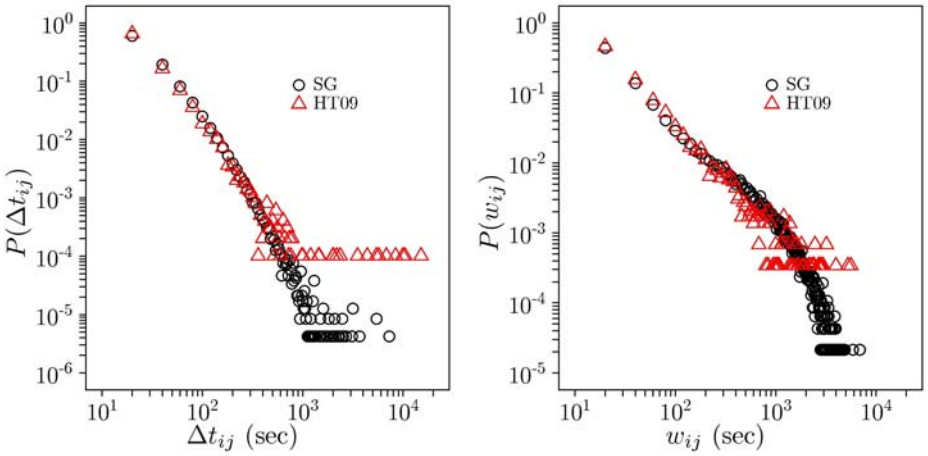

Figure 5: Left: contact duration distributions at HT09 (triangles) and at the SG (circles). Right: contact cumulative duration distributions at HT09 (triangles) and at the SG (circles).

We notice the broadness of both distributions which decay only slightly faster than a power law. Finally, we mention the strength distribution P(s), where s (node strength) is the total time an individual spends interacting with other individuals. As shown in Fig. 6 (right), s spans several orders of magnitude and P(s) is a monotonically decreasing quantity for the SG, with a plateau in the range from 2 to about 20 minutes. On the other hand (see the left diagram in Fig. 6) for the HT09 P(s) exhibits a peak for s=5 minutes.

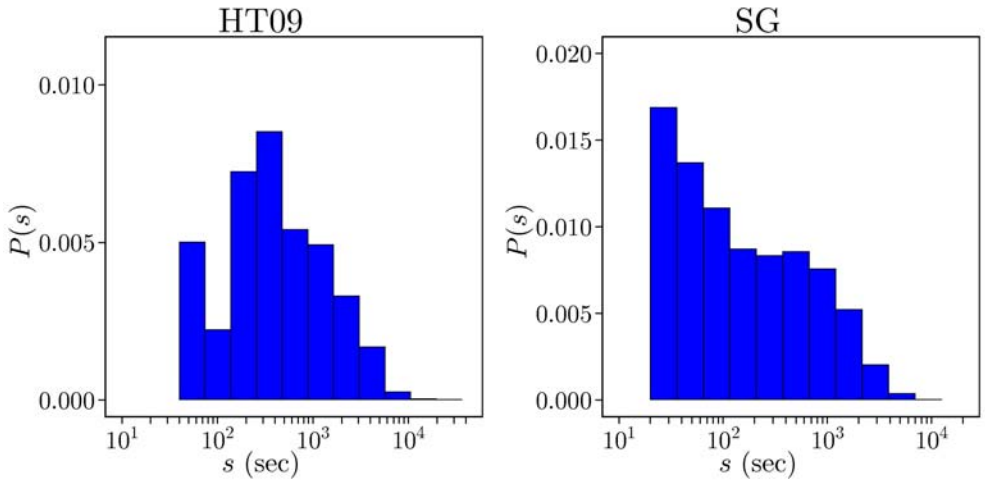

Figure 6: Strength distributions at HT09 (left) and at the SG (right).

The network visualizations in this section were produced using the igraph library [21]. Beside being visually appealing, they help researchers to intuitively grasp some of the most relevant (non necessarily only topological) features of the aggregated networks, which can be then be investigated quantitatively with the tools provided by WP3.

To conclude this section, we remark that the tools for network statistical analysis developed within WP3 allow for the investigation of many structural and longitudinal network properties, thus enabling the topological and dynamical characterization of the evolving network.

3.2 Elementary Dynamic Network Models (contribution by ColBud)

Network science still lacks a deep understanding of several properties of evolving networks. In particular, fundamental research on network resilience has mainly focused so far on classic network models for static networks, such as e.g. Erdos-Renyi and Barabasi-Albert. Network robustness has strong repercussions on the network's capability to sustain an information flow, hence an insight into the impact of node/edge dynamics on network resilience enhances the understanding of information spreading along a longitudinal network.

In order to investigate the resilience properties of dynamical networks and introduce appropriate statistical indicators, we created and studied Elementary Dynamic Network models (EDNs), the dynamic counterparts of the classic network models for static networks such as Erdos-Renyi (ER), Watts-Strogatz (WS) and Barabasi-Albert (BA) networks.

We describe briefly the EDNs used to carry out the numerical experiments

- ER1: Starting from an arbitrary network, in each time step, any non-existent edge is added with probability p and each existing edge is deleted with probability q.
- ER2: Starting from an arbitrary network, in each time step, k_1 existing edges are deleted (selected uniformly random from existing edges) and k_2 non-existing edges are added (selected uniformly random from non-existing edges).
- ER3 Starting from an arbitrary network, in each time step, k existing edges are rewired by moving one end of the link to another node.

The complete and systematic study of EDNs is ongoing work that has not been concluded yet and will be reported in the second project year. However, an innovative software tool (downloadable from the DynaNets home page) called Dyneta has been developed to generate and test EDNs. Dyneta is a program that generates networks (big, usually sparse graphs), runs an event on them and collects some statistics. An event can typically be adding or removing an edge such as in the ER models above. Dyneta is highly configurable - that means that it includes a wide range of network models and several events and statistics as well. The network models and events may have parameters too. Also, Dyneta has a plugin-based architecture and thus it can be extended easily. The Dyneta manual is available together with the software and the latter is in the public domain under Apache license. In order to understand the impact of one-time sampling an evolving network, we experimented with four EDNs, ER1, ER2 and ER3, as described above, plus a dynamic version of the classic Watts-Strogatz model.
In an applied study leading to the publication [3] we have investigated in detail the correlation of the average betweenness centrality (BW) with the network density (defined as the ratio of the actual number of links in the network to the maximum possible number of links in the network) while performing a series of operations on the network [3], in particular events of a destructive type. The importance of the BW of a node stems from its expressing the fraction of shortest paths that would be cut by the removal of that node, hence the average BW is a proxy for the expected number of shortest paths being affected by the deletion of a random node. Similarly, maximum BW points to the maximum damage that can occur by the removal of a single node. As a consequence, the average BW is an important structural indicator of network robustness.

Here we only report the results of the experiments we carried out with decreasing densities, in part, due to their immediate connections to existing works on robustness. In

network theory, robustness is understood as the resilience of networks in terms of connectivity and average path length, subject to the repeated removal of nodes (and links connected to them). We first successfully replicated the results in Ref. [22] regarding the random and degree-based removal of nodes. Then we introduced another removal scheme (a second “attack scenario”), i.e., the removal of high betweenness nodes. Our results show an initial decline in average betweennes in both Erdos-Renyi and Barabasi-Albert networks, as expected. However, and this is our new result, after a critical loss, the decline in average BW is followed by a new characteristic peak, as shown in Fig. 7, where each line is obtained by averaging the results on one hundred EDNs.

The investigation of EDNs and in particular of BW as a function of network size and density is of critical importance to understand information strpeading on dynamic networks, such as the spreading of a disease over a sexual contact network. Much ongoing work is in a pre-publication phase and will be reported accordingly.

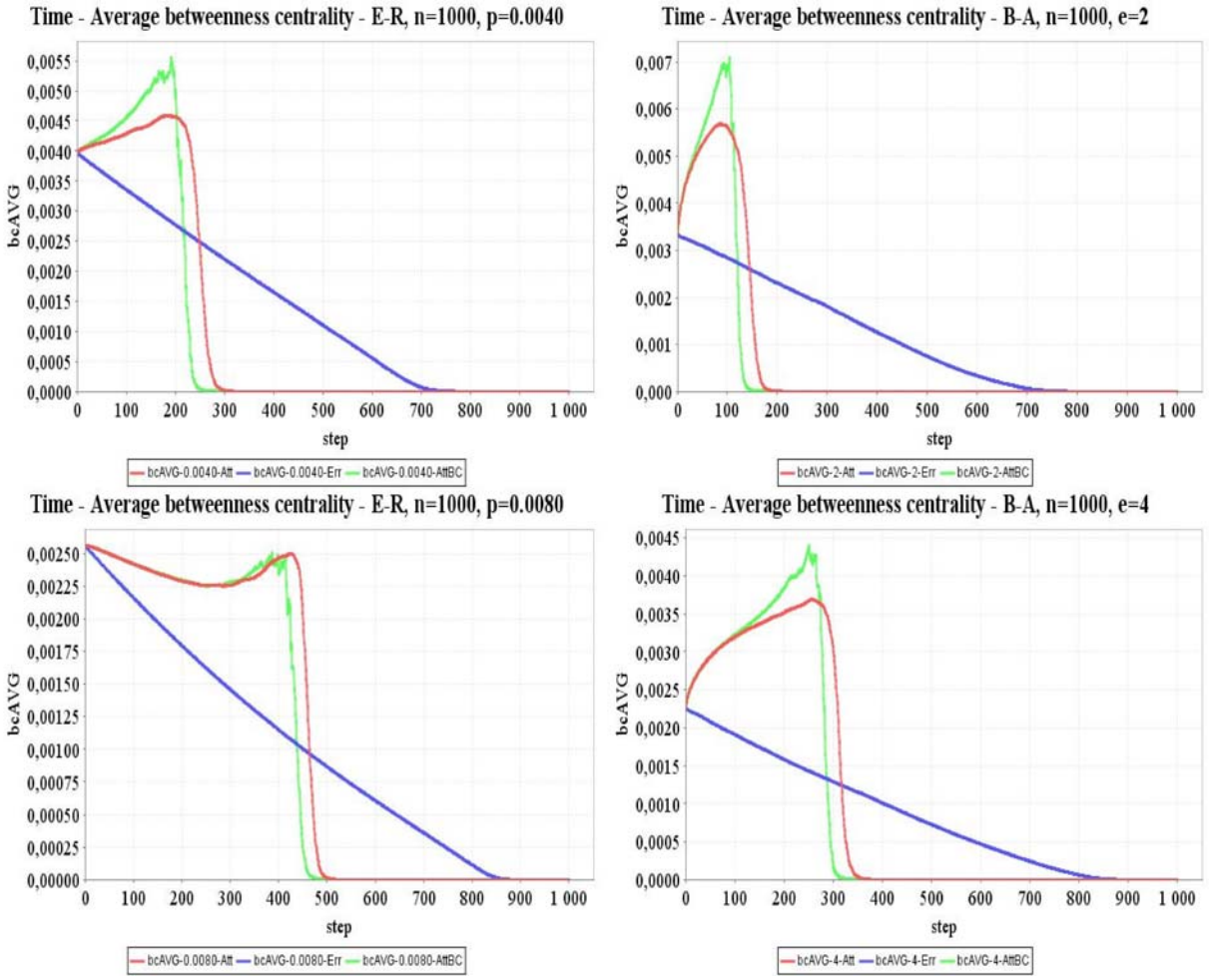

Figure 7: Average BW dynamics of ER and BA networks in time using various density lowering schemes. ER networks are on the left, BAs are on the right. In the first row the initial density is d = 0.004 (average degree is k = 4), in the second d = 0.008 (k = 8). Blue lines depict random node removals, reds are maximum degree-based, while greens maximum-BW based removals.

3.3 A Complex Network Simulator (contribution by UvA)

The simulation of dynamical processes unfolding upon a dynamical network is often a computationally daunting task as it may require the simulation of phenomena taking place on vastly different time scales. In addition, the complex networks that arise from such models will be so large that performance becomes an issue. Indeed multiscale, multiphysics systems are too complex for accurate analytical treatments, yet such systems arise everywhere from modeling the immune system and protein interaction to epidemic spread in a human population.

To address these problems we have developed a new tool, the Simulator for Efficient Evolution on Complex Networks (SEECN [23]), an expressive simulator of complex systems. Emphasis was given to the optimization and fine-tuning of the simulator which aims at showing the feasibility of a network science approach to problems which have been traditionally considered too demanding for accurate numerical simulation. In SEECN, a complex network represents the system where the nodes and edges have specified properties which dictate the dynamics of the network over time. It features the capability of integrating the information from different time scales while confirming the computational feasibility of agent-based modeling combined with complex networks. As a case study to validate SEECN, we developed a detailed model of HIV spread among men who have sex with men and serves to show the simulator's expressiveness and to evaluate its performance. The investigation of the spreading of HIV and its drug resistance represents an excellent stress test for SEECN since requires a holistic approach of various dynamics at multiple spatiotemporal scales.

Within DynaNets, SEECN's functionality has been extended for hierarchically modular networks, necessitating changes to the code for parallelism as well. Also, nodes may now have any number of states that may take on many values, such as integer values. As a consequence, the specification of model parameters has been generalized to support functions in order to be able to handle such numerous possible states: as an illustration, specifying infection probabilities is in general an S-by-S-by-S matrix of probabilities, where S is the number of possible node states. In this case, a function could exploit regularity in the parameters and be much more concise. Lastly, an important step was to support node removal independent from node addition, necessitating a broad restructuring of the optimizations for cache efficiency and consequently in parallelism.

The epidemic model simulated with SEECN is fairly complex and represents the HIV infection network in a population of homosexual men, therefore each node has the same type (male). This model assumes a hierarchical network with power-law exponent 1.6 (i.e. the probability that a homosexual has n sexual partners in a 3-month time window decays like $P(n) \sim n^{-1.6}$), and classifies nodes as healthy, acute, (un)treated asymptomatic, or (un)treated AIDS.

We assume that an untreated HIV patient has an expected progression time into AIDS of thirteen years, and a slightly lower median. The expected progression time of a treated HIV patient into AIDS is taken to be about twenty-two years. Acute HIV lasts for approximately three months, which we take as the length of one time step. Treatment

uptake is typically modeled as a percentage of untreated patients per quarter. We calculate the uptake to ensure that about 70% of a node's time of infection is under treatment, and assume that 30% of the treatments fail. Various drug treatments may be introduced at different times, but in our simulations we model only one treatment type that is available at time step zero, and the use of which gradually increases. We further assume that about 60% of untreated infected nodes is actually diagnosed and aware of their status, which results in 25% less risky behaviours. Nodes in the acute HIV stage are taken to be much more infectious due to a peak of viral body count. Dynamics and parameters also include condom use, scale-free distribution of number of partnerships including community structure, partnership duration, and distinction of steady and casual partnerships. For a detailed list of the parametrization of node and edge dynamics in the HIV infection network, see appendix A of Ref. [24]. We have found remarkable resemblance to historical data (from Amsterdam Cohort Study and San Francisco cohort studies).

The added value of SEECN in the framework of the software tools provided by WP3 is that it offers a high-performance tool for running simulations of and on complex networks. SEECN is indeed a self-integrated tool (it created its own initial network from a set of parameters and it evolves the network calculating only a limited subset of the potential quantities of interest) for performance reasons, but this has repercussions on its flexibility and ease of use. For the foreseeable future we thus expect SEECN to be used when computational performance is an issue, and the tools it replaces to be used in general.

3.4 Multiscale framework within the CoSMo Toolkit (contribution by ENS)

The demanding numerical optimizations implemented in SEECN unavoidably lead to a partial lack of flexibility of the computational engine. Due to the importance of multiscale modeling in network science, we introduced the CoSMo toolkit as a flexible multiscale computational framework. The CoSMo toolkit is a computational platform for the rapid prototyping and implementation of multiscale models for dynamical processes unfolding upon a longitudinal network.

As a proof of concept, among many possible options, we developed an algorithm simulating the spread of a disease (e.g. influenza) in a novel, multi-scale fashion. The algorithm has been developed bearing in mind the possibility of future extensions to include the ever-increasing availability of data about human activities.

To this aim, we resort to a hierarchical network comprising different spatial and temporal scales, hereafter referred to as levels. So far we have implemented human dynamics at

the individual level within a subpopulation representing a city, and the migration flow of individual between cities (city level). Human dynamics at the individual level is simulated by resorting to an agent-based model where every individual stands for a node characterized by an infection status (modeled according to an SIR model), a viral load, its mobility properties and its relations with other individuals. In the network at the city level, instead, every node stands for a city characterized by properties such as seasonality, natality/mortality rates and a migration rate establishing a link between different cities. The algorithm allows for the future integration of real data for mobility and node features for specific applications to the dynamics of disease spreading.

3.5 Agent based model for social interactions (contribution by ISI)

Within the scope of WP3 we developed, as a specific application, a network model aimed at capturing the main features of the conversational dynamics of individuals interacting in a limited space, such as the case of a conference venue. The model cannot be extended trivially to other kinds of social interaction.

The statistical analysis of networks of human interactions carried out in section 3.1 highlights the existence of broad contact duration and link weight distributions. We developed an agent-based model for human dynamics [6], which characterizes the system with a limited number of statistical parameters. The model we propose considers a fixed population of N agents interacting in a limited space. Therefore we neglect the spatial dispersion of the agents and we assume a well mixing dynamics. Each agent can either be isolated or belong to a group with other agents, and the groups define an instantaneous contact network. During the dynamics, agents can join other agents or on the contrary leave the group they belong to. Hence the state every agent is characterized by only two parameters, namely the number p_i of other agents with which it is in contact (i.e. its degree in the network) and the time t_i at which p_i last evolved. The final ingredients of the model are the probabilities that an agent changes its state. For an isolated agent, changing its state amount to joining a group, whereas an agent within a group can either leave the group or introduce an isolated agent to the group.

The model enforces a simple rule for human dynamics which can be stated as “the longer an agent interacts with a group, the less it is likely to leave the group; the more the agent is isolated the less likely it is to interact with a group”. Mathematically this

translates into the probability for agent to change its state at time t (in one of the ways described above) to be a decaying function of $t - t_i$ i.e. of the time interval from the last time p_i evolved. One of the consequences of this assumption is that the lifetime of large groups tends to be shorter than that of smaller groups. Indeed, as a first approximation, these decision correspond to independent events, hence groups with more agent become less stable. In Fig. 10 we plot the results of some numerical experiments, namely the distribution of contact durations between two agents (a), time intervals between the beginnings of successive contacts of an agent A with two different agents B and C (a measure of the ability of an agent to efficiently spread information) (b) and the triangle duration probability distribution (c) for two different values of the parameter b_1, which parametrizes the tendency of an agent to leave an existing group [6]. The numerical calculations qualitatively and quantitatively agree with analytical approximations [6], since they both predict fat-tailed contact/interval duration distribution for realistic choices of b_0, b_1 and λ as shown in Fig.8 and in agreement with what observed empirically [25].

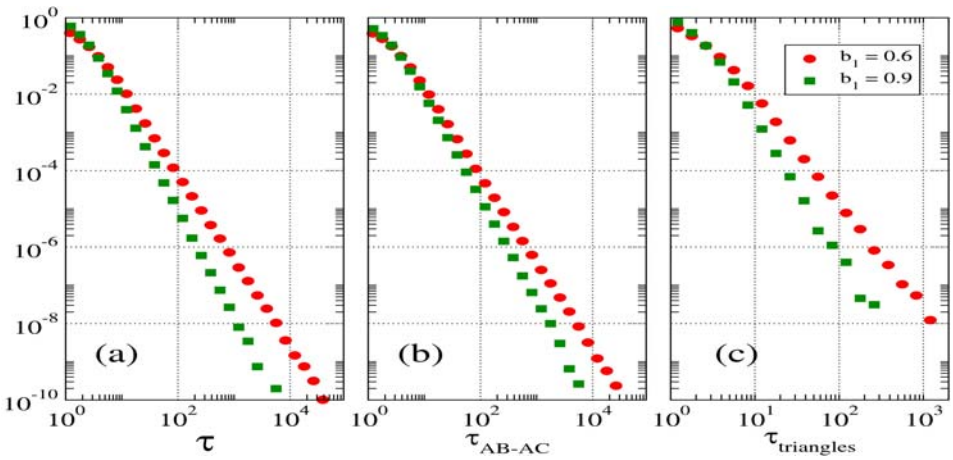

Figure 8: Distributions of (a) duration of a contact between two agents; (b) time intervals between the beginnings of successive contacts of an agent A with two different agents B and C; (c) duration of a triangle.

4 Implementation and Integration of Software Tools (joint effort by all partners)

The integration between the various tools is realized at the input/output level as sketched in Fig. 9. The tools for statistical analysis are able to post-process the output both of the

EDNs models (consisting of SONIA/Pajek files) and of the CoSMo toolkit (a time-dependent edgelist). For the EDNs models, the input format being homogenous with the output format, the network analysis scripts developed by ISI can also be deployed to generate/analyze the initial state. At the present the CoSMo toolkit is unable to read a network from a data file, hence there can be no integration at the input level with the statistical tools, but we plan to add this functionality in the near future.

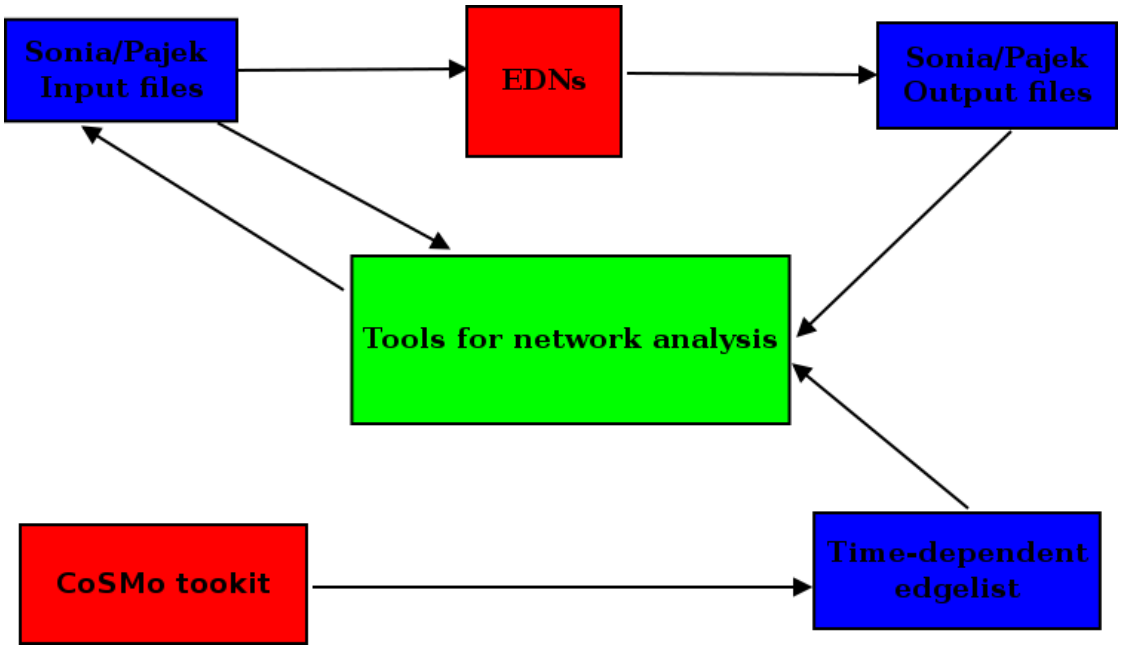

Figure 9: Integration of the software tools provided by WP3.

There is no integration with SEECN which is essentially a self-contained tool due to the extreme nature of its computational optimization and the large scale of the problems it aims to address. As explained in the following, it would be unfeasible to store the initial states needed by SEECN in a set of files or to take statistics of more than a few carefully selected quantities of interest.

Table 2 shows an overview of the implementation details of the tool contributed to WP3 complemented by a list of the external libraries they rely upon. We notice that there are only two libraries deployed for the heavy numerical work of network simulation and analysis, namely the igraph library (used both by the tools for statistical analysis and by EDNs models) and the Boost library [28] (shared by the CoSMo toolkit and by SEECN). Both libraries are freely available and supported by an active community.

All the software tools for the statistical analysis of networks of human interactions were developed either using Python or the R language for statistical analysis. Both scripting languages are widely spread in the scientific community and provide a wide choice of high quality packages for statistics and network analysis, which are typically a thin layer over modules written in C/C++ or Fortran for performance reasons. We sketch the

flowchart that we follow when applying the scripts to the raw data in Fig. 10. The raw network data is a simple text file formatted as a time-dependent edgelist. We note the presence of two separate branches in the flowchart since, as explained in Section 3.3, calculating contact durations/intervals is a task independent from generating the aggregated networks. Since detecting contact durations/intervals is operationally equivalent to performing a series of array operations, we resort, due to its speed and reliability, to the Python scientific module NumPy [26]. The calculated contact distributions are themselves saved as text files storing lists of integer times (since every contact/interval is bound to be an integer multiple of a the 20 sec time slice used to sample the human interactions).

Table 2: Overview of the developed software tools and the external libraries used in their implementation.

ISI	UvA	ColBud	ENS
Tools for network analysis	SEECN	EDNs	CoSMo
Python and R scripts for network statistical analysis and modeling.	C/C++ simulation engine.	C codes for robustness measures of dynamical networks.	CoSMo toolkit as a computational platform.
NumPy, igraph, ggplot2.	Boost, MPICH2	igraph (C interface)	Cmake, Boost, libxml2, Tulip, Qt.

This output is in turn fed to other R scripts carrying out further post-processing (such as e.g. binning the distributions) and statistical visualization, thus outputting new text files with high-level quantities of interest complemented by their statistical visualization (pdf figures). Statistical visualization relies on the ggplot2 [27] R library, due to its high degree of flexibility and the ease with which it can be deployed to produce publication-quality figures.

In the other branch of the flowchart, the same initial raw data is fed to another R script making ample use of the igraph library for network analysis and visualization. The script breaks the recorded interactions into 24-hour time intervals along which it generates the corresponding aggregated networks. These networks are naturally weighted by the link cumulative durations. Such networks are stored using the Pajek format [33] provided by the igraph library. The choice of this particular format for storing networks is by no means

compulsory, but we found it very convenient as it stores all the available network information in a single portable text file. Furthermore, Pajek is a widespread tool for network visualization hence saving networks as Pajek files may ease the sharing of information and results among researchers.

The generated aggregated networks are in turn fed to other R scripts for further data analysis. The ggplot2 library is again used for statistical visualization, whereas network static visualization can be achieved resorting to the graphic capabilities of the igraph libraries. We note that the final output of both branches of the flowchart in Fig. 10 consists of pdf and text files only, thus being highly portable to different platforms.

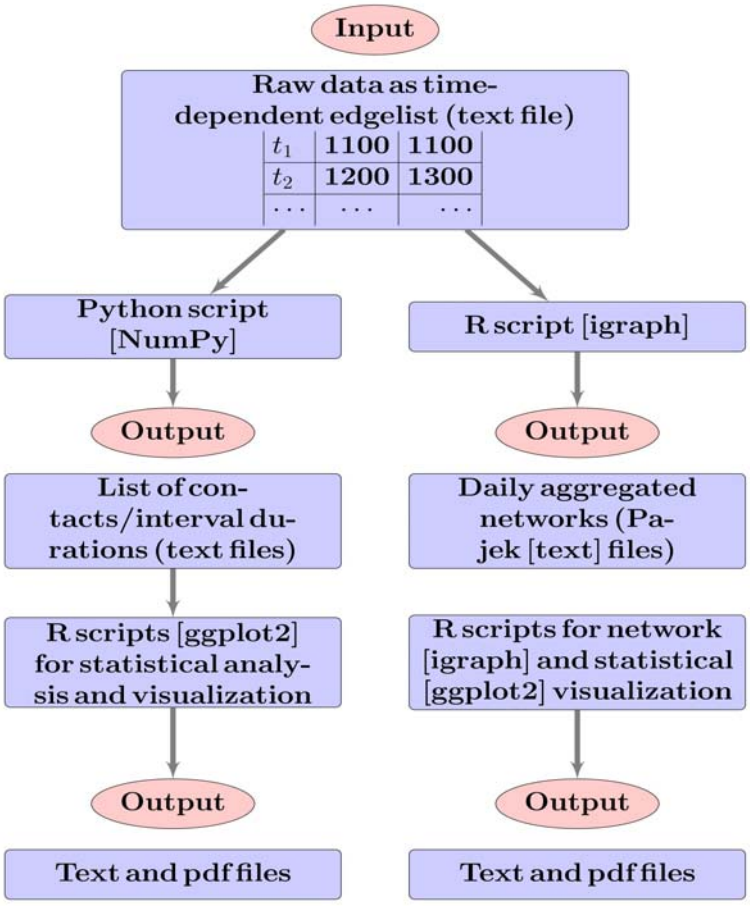

Figure 10: Flowchart of the statistical analysis of network data.

We finally remark that all the mentioned libraries/scientific modules (namely NumPy, igraph and ggplot2) are freely available as open source software.

The input files for the CoSMo toolkit are xml files where one defines an instance of the network model consisting of the corresponding type of graph and its parameters (e.g random, scale free, Erdos-Renyi, small world, etc...). As of now CoSMo does not allow the user to directly load an empirical network from an input file. On the other hand, CoSMo is fully capable of producing a time-dependent edge list hence the ouput of its computational models can be analysized with the postprocessing tools developed by ISI.

Finally, most of the code for the CoSMo toolkit is written in C++, though some wrapper functions are coded in Java. Furthermore, CoSMo also relies on third party libraries, namely

- Cmake for the cross platform compilation (CoSMo compiles on Linux, Windows and Mac) [29],
- Boost for The Boost Graph Library (BGL) [28].
- libxml2 for parsing xml input and output file [30],
- Tulip for graphic visualization [31].
- Qt for the Graphical User Interface [32].

In the case of SEECN, the network is generated stochastically by SEECN itself and no input file is needed. SEECN saves only some selected high-level statistics as standard CVS files, but not the network state as this may easily lead to disproportionally large files when dealing with networks consisting of millions of nodes. The complex network simulator deployed for the HIV, SEECN is a single C++/MPI program relying on the open source library Boost [28] and is optimized for both single-core and multi-core performance. We allocate one process per compute node (homogenizing communication overhead) and use TCP/IP over Infiniband. We implement SEECN in C++, parallelize it with MPI, and compile it with GCC 4.1.2 and MPICH2 1.0.8. SEECN's code is not hand tuned and uses the C++ standard template library sorting algorithm.

The tools developed by ColBud to investigate the robustness of EDNs are written in C and produce their output networks in two possible formats: a series of Pajek (.net) files or a single file using the so called SONIA (.son) format. The former file type is natively supported by the igraph library hence it can be easily fed to the tools for network statistical analysis provided by WP3. The latter is an extension and slight modification of the former, following a column based, rather than a token based concept and, more importantly, allowing for the time stamping of the nodes and edges. The igraph library does not support this format natively, but it can be extended to add this functionality.

Overall, the input/output of the EDNs definitely lends itself to the analysis by means of the statistical tools in WP3.

5 Performance report (contribution by all partners)

A straightforward comparison of the performance of the different tools contributed to WP3 is impossible due to the heterogeneity both of the tasks they carry out (network analysis versus simulation of network dynamics) and of the platforms where they run (desktop PC, multi core workstation, cluster). Nevertheless, Table 3 provides an overview of the benchmarks and running time for each computational tool.

The statistical analysis of dynamical networks can be carried out quite efficiently with the tools provided by D3.1. The analysis of the networks of social interactions has been performed on a quad core Z600 workstation with 8 Gb of RAM memory. The software tools for network statistical analysis do not parallelize the computations, but as explained in the previous section, the contact statistics and the aggregated network analysis can be carried out simultaneously.

The memory footprint of these applications is very light (at most 13-14% of the total RAM memory is used and only when taking the contact statistics). Numerically, the most challenging part of the post-processing is the first step performed on the raw data. The network analysis can be completed within a matter of a few minutes as it reduces to the analysis of many small networks, each consisting at most of a few hundred nodes, aggregated on a 24-hour basis. Indeed even the large dataset collected at the SG recording the interactions of tens of thousands visitors ends up broken into about 80 daily aggregated networks each consisting of a few hundred nodes. We point out that the aggregation time can be given as an input parameter hence one is able to aggregate the network on longer/shorter periods if needed. The contact duration/interval analysis takes longer as in this case the whole dataset is read and the occurrences of a given tag ID are looked for throughout the whole input data.

The time-dependent edge list proves to be a very convenient format as it is very close to the internal representation of the dataset by the R/Python scripts, where contacts and networks are actually represented by multidimensional arrays.

Finally, we mention that the software tools for network analysis make extensive use of the igraph library [23] which has been reported to be able to handle network sizes of

several hundred thousands of nodes or more. As a consequence, we predict the scalability of the software tools developed for the analysis of networks of human contacts up to networks consisting of about 100.000 nodes at least. Furthermore, we remark that we used the same hardware to generate all the network visualizations in this report.

The network plots are generated once again resorting to the igraph library [23] within a few tens of seconds at most.

As a benchmark, in table 3 we provide the running time for analyzing on the aforementioned HT09 and Dublin dataset on a quad core HP Z600 workstation. For both cases, we also specify the length (i.e. number of rows) of the time-dependent edge list.

Table 3: overview of the benchmarks for the contributed tools.

Platform	Code	Case Studies	Execution time/Benchmark Details		
Quad core workstation HP Z600, 8Gb of RAM, 2.13 GHz.	Tools for network analysis (R/Python scripts)	HT09 (length of edgelist = 350.000)	2 minutes for contact analysis	2 hours for contact analysis	
		SG (length of edgelist = 2.000.000)	2 minutes for network analysis	5 minutes for network analysis	
Linux cluster of 680 dual-core Intel Xeon nodes at 3.4 GHz.	SEECN (C++)	1.000.000 nodes, 100 time steps	2 minutes		
PC, Intel Core Duo T5600 1833 MHzCPU and 2 Gb of RAM	EDNs (C)	50000 iterations, 100 runs	29 days		
Quad core Dell Precision workstation T7500, 4Gb of RAM,	CoSMo toolkit (C++, Java)	Generation of static network (10.000 nodes)	Simulation	Memory	Time
			Edges =0	4.8 Mb	0.06s
			Edges=100	4.9 Mb	0.25s

2.27 GHz.			Edges =1.000	5.2 Mb	1.9s
			Edges =10.000	9.2 Mb	5.2s
			Edges =20.000	13.5 Mb	37.5s
			Edges =40.000	22.6 Mb	75s
			Edges =80.000	40 Mb	150s
			Edges =160.000	76 Mb	300s
		Network rewiring (10.000 nodes)	Simulation	Memory	Time
			Edges=5.000	69 Mb	210 s
			Edges=10.000	108 Mb	1260 s

We point out that even on a large datasets (consisting of thousands of monitored individuals establishing hundreds of thousands of contacts) the analysis can be performed in a very affordable time.

In the case of SEECN, the numerical experiments described in this deliverable were run on a Linux cluster of 680 dual-core Intel Xeon nodes at 3.4 GHz. The required computational time depends of course on the size and complexity of the desired agent-based simulation. The sequential computing times of generating the network (G) and performing all other dynamics in a single time step (D) are shown, on a logarithmic scale, in Fig. 11. All data points are averages of five runs, and each run had 100 time steps of dynamics (standard error is too small to show).

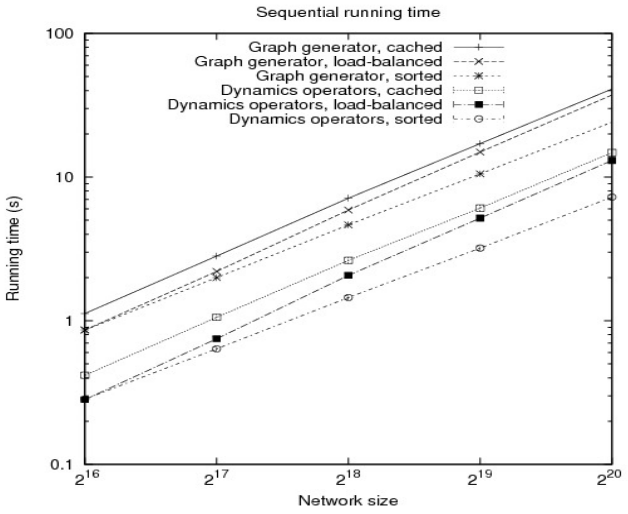

Figure 11: sequential running time of a single iteration of the HIV network model in SEECN as a function of network size.

The computation time roughly scales linearly with the number of edges, which translates, for a scale-free network, into a superlinear dependence on the number of nodes (i.e. number of individuals). Despite the superlinear growth of the computation time with the number of nodes, we report as a benchmark that a detailed simulation of HIV among one million persons in a hierarchical and scale-free network over 25 years (100 time steps) takes two minutes using 16 processes. As a consequence, SEECN shows the feasibility of agent-based simulations of HIV spreading on a realistic network with an accurately chosen parameterization.

As to the EDNs experiments, we point out that the numerical experiments we ran were not computationally expensive, as the basic models that we experimented with had a size N=100. However, considering the size of real networks (where 100 nodes count as extremely small) and the possible number of connections that increases quadratically with the number of nodes, simulations of EDNs may require a considerable computational effort. Especially, as for a full experimental treatment, the same simulation has to be run several times with the same parameter combinations (but with different random number seeds) in order to assess the robustness of our findings.
Moreover, we also intend to do sensitivity analysis (i.e., the replication of our measurements with different parameter combinations). All these together will amount to a significant computational effort, but all within the controllable range. During our

preliminary experiments so far, we have been studying networks of 100 nodes (N=100). In case of the ER1 model, we have started from an empty network (initial density = 0), and added each non-existent link with probability p=0.0002, and deleted each existing link with probability q=0.001 in each iteration. For the ER2 model, we have also started from an empty network (initial density of 0), but experimented with various parameter combinations: $k_1 = k_2 = 1; k_1 = 2$, $k_2 = 0$ and $k_1 = 4$, $k_2 = 2$. In case of the ER3 model, we have experimented with various initial networks (ER networks with density 0.0404 and 0.9091, respectively, and with Watts-Strogatz network with neighborhood reach 2 and 45, respectively), keeping parameter k=1.
In case of all experiments, we have completed 50000 iterations and with each setting we have taken 100 independent samples (run the experiments with 100 different pseudo random number seeds). For this purpose, we have used a standard office computer a dual core Intel Core Duo T5600 1833 MHz CPU and 2 Gb of RAM. On this setup the experiments took 29 days of total net execution time.

For the CoSMo toolkit, we report the in Table 3 time it takes CoSMo to generate a random network with 10000 nodes and a varying number of edges. Time roughly scales linearly with the number of edges.
In Table 3 we also report the time it takes to rewire 100 times a network consisting of N=10000 nodes connected by E=5000,10000 edges. Here we notice a non-linear dependence on the number of edges of the computation time, though the memory footprint scales roughly linearly. In both cases, the simulations were run on a quad core Dell Precision workstation equipped with 4Gb of RAM memory and a 2.27GHz CPU.

All the computational tools described in this deliverable were, whenever possible, stress-tested on either real or at least realistically-sized networks. Computationally less demanding tasks, such as longitudinal network analysis (contribution by ISI), investigation of subtle theoretical aspects of fundamental network models (EDNs by ColBud) and the fast ab initio development and implementation of new multiscale models (contribution by ENS) can be carried out on any up-to-date workstation, whereas we provide an already optimized complex network simulation (SEECN by UvA) to carry out extremely demanding dynamical network simulations on a cluster.

6 References

[1] See the Infectious website at http://sciencegallery.com/content/science-gallery-2009-infectious .

[2] See the Hypertext 2009 website at http://www.ht2009.org/ .

[3] L. Gulyas, G. Horwath, T. Cseri, Z. Szakolczy, and G. Kampis, *Betweenness centrality dynamics in networks of changing density*, Proceedings of the 19th International Symposium on Mathematical Theory and System , Budapest, Hungary, 5-9 July (2010).

[4] See http://www.amsterdamcohortstudies.org/.

[5] M. H. Katz, S. K. Schwarcz, T. A. Kellogg, J.D. Klausner, J. W. Dilley, S. Gibson, and W. McFarland, *Impact of Highly active Antiretroviral Treatment on HIV Seroincidence Among Men who Have Sex with Men: San Francisco,* J Public Health 92, 388 (2002).

[6] J. Stehlé, G. Bianconi, and A. Barrat, *Dynamical and bursty interactions in social networks,* Phys. Rev. E 81, 035101 (2010).

[7] P. Hui, J. Crowcroft, and E. Yoneki, *BUBBLE Rap: Social-based Forwarding in Delay Tolerant Networks*, Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, 244 - 251 (2005).

[8] N. Eagle, and A. Pentland, *Reality Mining: sensing complex social systems,* Personal and Ubiquitous Computing 10, 255-268 (2006).

[9] E. O'Neill, E. Kostakos, V. Kindberg, A. Fatah Schlek, A. Penn, D. Stanton Fraser, and T. Jones, *Lecture Notes in Computer Science* 4206, 315, Springer (2006).

[10] A. Pentland, *Honest Signals: how they shape our world*, MIT Press, Cambridge MA, (2008).

[11] A. Clauset, and N. Eagle, *Persistence and periodicity in a dynamic proximity network*, DIMACS Workshop on Computational Methods for Dynamic Interaction Networks (2007).

[12] http://www.sociopatterns.org .

[13] C. Cattuto, W. Van der Broeck, A. Barrat, V. Colizza, J.F. Pinton, and A. Vespignani., *Dynamics of person-to-person interactions from distributed RFID sensor networks*, PLoS ONE 5(7), e11596 (2010).

[14] H. Alani, M. Szomsor, C. Cattuto, W. Van der Broeck, G, Correndo, and A. Barrat. *Live Social Semantics*, 8th International Semantic Web Conference ISWC2009, LNCS 5823, 698-714 (2009), http://dx.doi.org/10.1007/978-3-642-04930-9_44 .

[15] W. Van den Broeck, C. Cattuto, A. Barrat, M. Szomsor, G. Correndo, H. Alani. *The Live Social Semantics application: a platform for integrating face-to-face presence with*

online social networking, First International Workshop on Communication, Collaboration and Social Networking in Pervasive Computing Environments (PerCol), 226 (2010).

[16] Special issue of Science on Complex networks and systems. Science 325, 357-504 (2009).

[17] S. N. Dorogovtsev, and J.F.F. Mendes, *Evolution of networks: From biological nets to the Internet and WWW,* Oxford University Press, Oxford, (2003). M. E. J. Newman, *The structure and function of complex networks*. SIAM Review 45:167 (2003). R. Pastor-Satorras, and A. Vespignani, *Evolution and structure of the Internet: A statistical physics approach*, Cambridge University Press, Cambridge (2004). G. Caldarelli, *Scale-Free Networks,* Oxford University Press, Oxford (2007). A. Barrat, M. Barthélemy, and A. Vespignani, *Dynamical processes on complex networks*, Cambridge University Press, Cambridge (2008).

[18] D. J. Watts, *A twenty-first century science,* Nature 445, 489 (2007).

[19] A. Wasserman, and K. Faust, *Social Network Analysis: Methods and applications*, Cambridge University Press, Cambridge (1994).

[20] S. Maslov, K. Sneppen, and A. Zaliznyak, *Detection of Topological Patterns in Complex Networks: Correlation Profile of the Internet,* Physica A 333, 529 (2004).

[21] G. Csardi, and T. Nepusz, *The igraph software package for complex network research*, InterJournal Complex Systems, 1695 (2006).

[22] R. Albert, H. Jeong, and A. L. Barabasi, *Error and attack tolerance of complex networks*, Nature 406, 378 (2000).

[23] SEECN available at http://staff.science.uva.nl/~rquax/seecn.html .

[24] R. Quax, *Modeling and simulating the propagation of infectious diseases using complex networks*, M.Sc. thesis available at http://staff.science.uva.nl/~rquax/public/quax_rick_200808_mast.pdf .

[25] A. Barrat, C. Cattuto, V. Colizza, J. F. Pinton, W. Van der Broeck, and Alessandro Vespignani, *High resolution dynamical mapping of social interactions with active RFID,* e-print arXiv:0811.4170.

[26] T. E. Oliphant, *Python for Scientific Computing*, Computing in Science & Engineering, vol. 9, no. 3, May/June 2007, pp. 10-20 and see also http://numpy.scipy.org/ .

[27] H. Wickham, *ggplot2: elegant graphics for data analysis*, Springer, New York (2009).

[28] http://www.boost.org/ .

[29] http://www.cmake.org/ .

[30] http://xmlsoft.org/ .

[31] http://tulip.labri.fr/TulipDrupal/ .

[32] http://qt.nokia.com/ .

[33] V. Batagelj, and A. Mrvar: Pajek – *Program for Large Network Analysis,* available at http://vlado.fmf.uni-lj.si/pub/networks/pajek/ .