

# Визуализация больших графов

Фадеева Анастасия, 371

# Актуальность

- Биоинформатика
- Анализ социальных сетей
- Маркетинг
- Социология

# Проблемы визуализации больших графов

- Производительность
- Различимость элементов
- Понимание (можно отобразить общую структуру, но не внутренние взаимосвязи)

# Мотивация

Разные методы  
раскладки



Различные  
внешние  
представления  
графа



Различное  
восприятие  
пользователем

# Методы раскладки графов

- Использование физических аналогий (force-directed)
- Понижение размерности (dimension reduction)
- Спектральные методы (Spectral method)
- Многоуровневая раскладка (Multi-Level)

# Выборка графов

- Random Node
- Random Edge
- Random Edge-Node
- Random Walk
- Random Jump
- Forest Fire

# Визуальное восприятие

**Что влияет на визуальное восприятие:**

- Вершины высокой степени
- Краевые вершины - низкая степень, находятся не в центре графа
- Пограничные вершины - между двумя и более кластерами
- Качество кластера
- Область покрытия

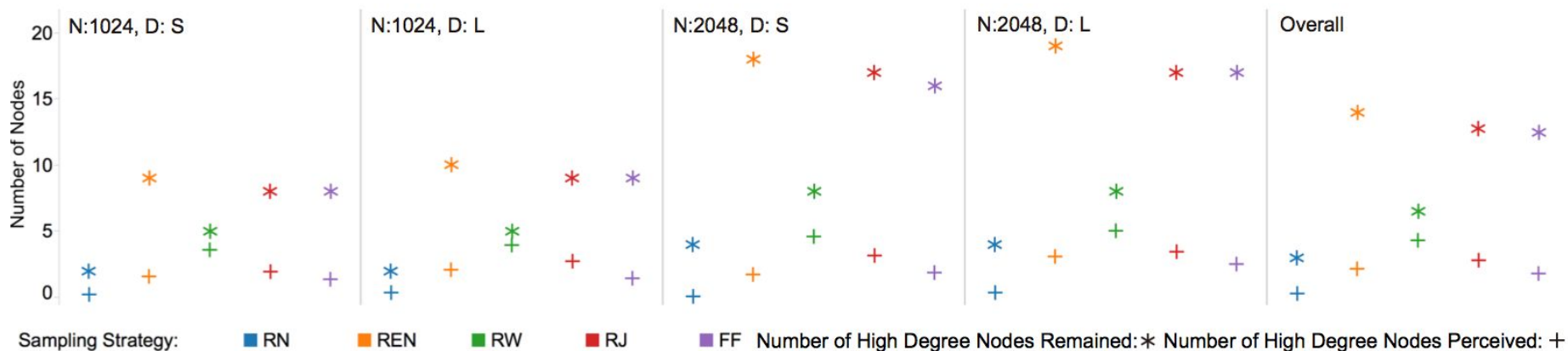
# Данные

## Генераторы графов:

- **Barabási-Albert** - гарантирует, что в сгенерированном графе будут вершины высокой степени
  - **Параметры:**
    - число вершин
    - среднее значение степени для вершин высокой степени
- **Sah. et al.**
  - **Параметры:**
    - число кластеров
    - значение модулярности (модулярность - оценка качества разбиения графа на “сообщества”)

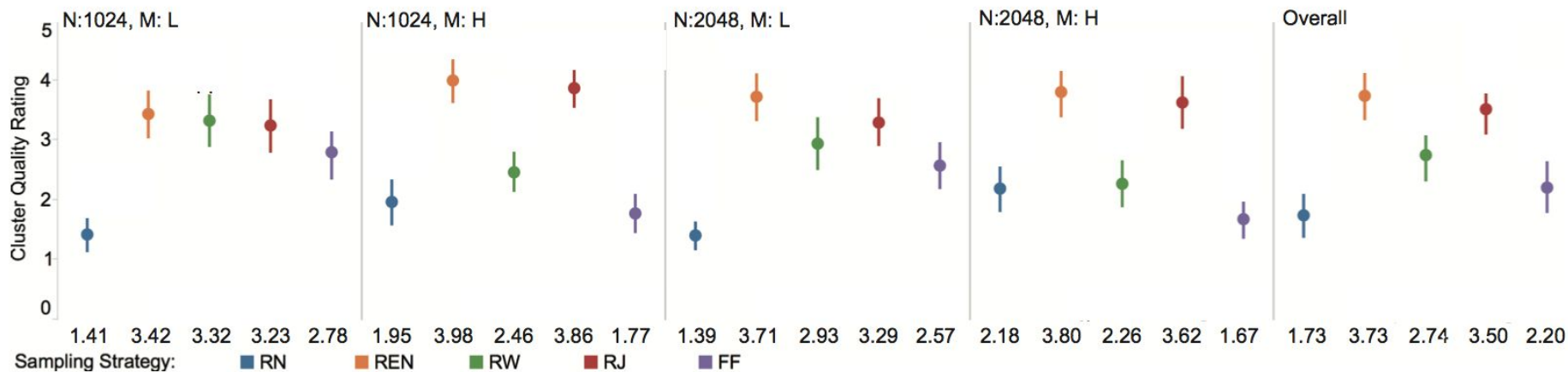


# Алгоритмы выборки и вершины высокой степени



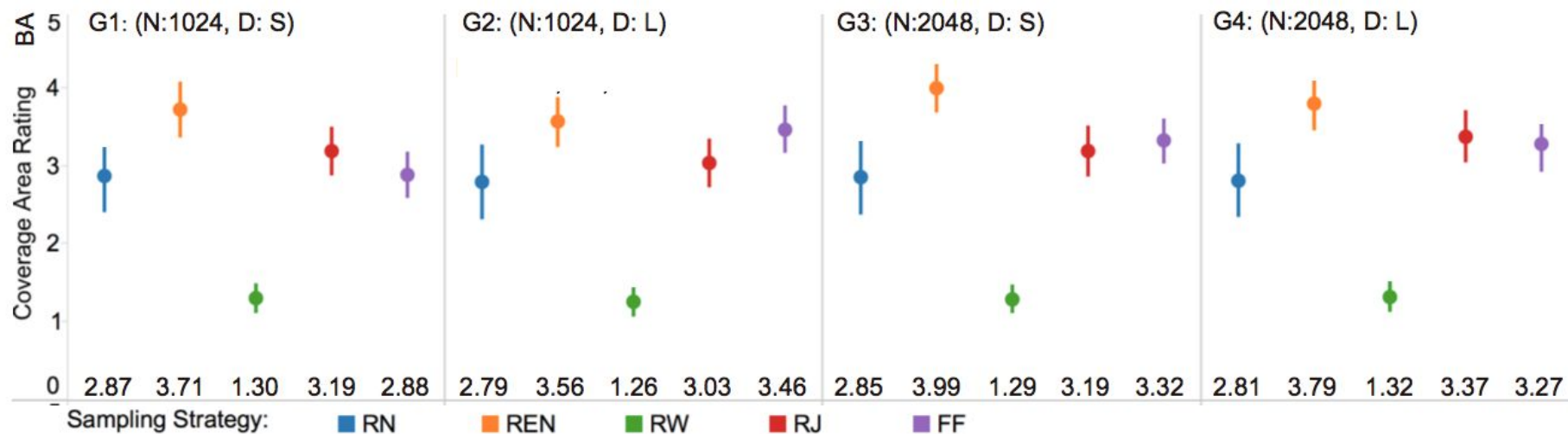
- Легко “распознать” вершины высокой степени в выборе графа, полученной алгоритмом RW
- Достаточно сложно “распознать” высокой степени в сэмпле графа, полученном RN

# Алгоритмы выборки и качество кластеров



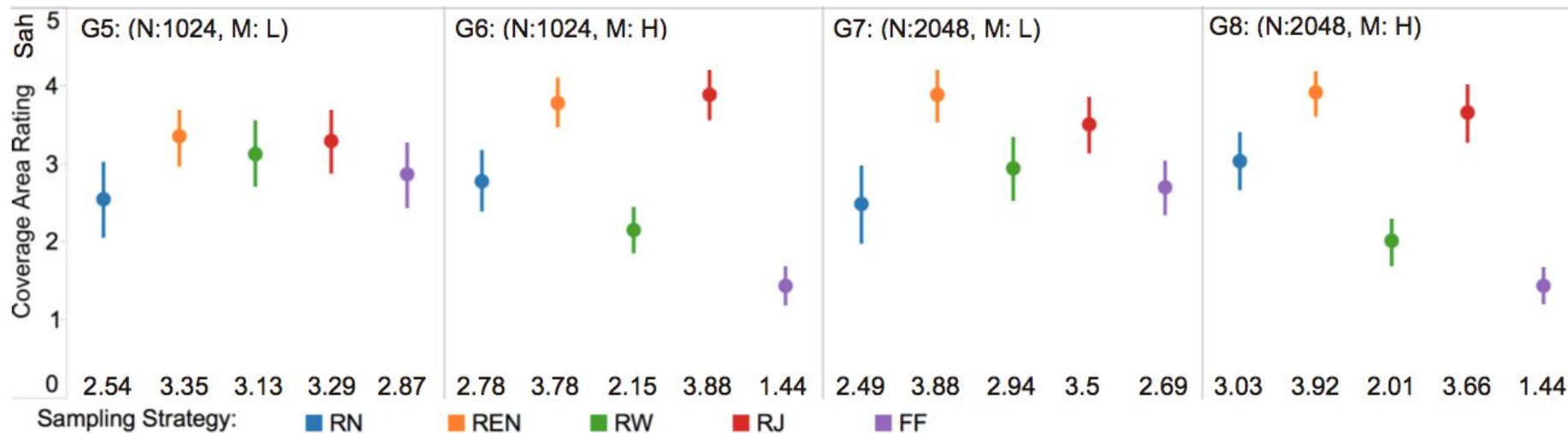
- REN и RJ лучше всех исследуемых алгоритмов раскладки сохраняют Cluster Quality
- Результаты относительно метрики Cluster Quality для алгоритмов RW и FF зависят от модулярности и не зависят от размера графа

# Алгоритмы выборки и область покрытия



- REN и RJ имеют наибольшую область покрытия
- Результаты RW и FF зависят от свойств графа

# Алгоритмы выборки и область покрытия



- REN и RJ имеют наибольшую область покрытия
- Результаты RW и FF зависят от свойств графа

# Рекомендации

- REN & RJ - для сохранения общей структуры и качества кластеров
- RW - сохранения восприятия вершин высокой степени
- Избегать RN
- RW & FF - чувствительны к модулярности

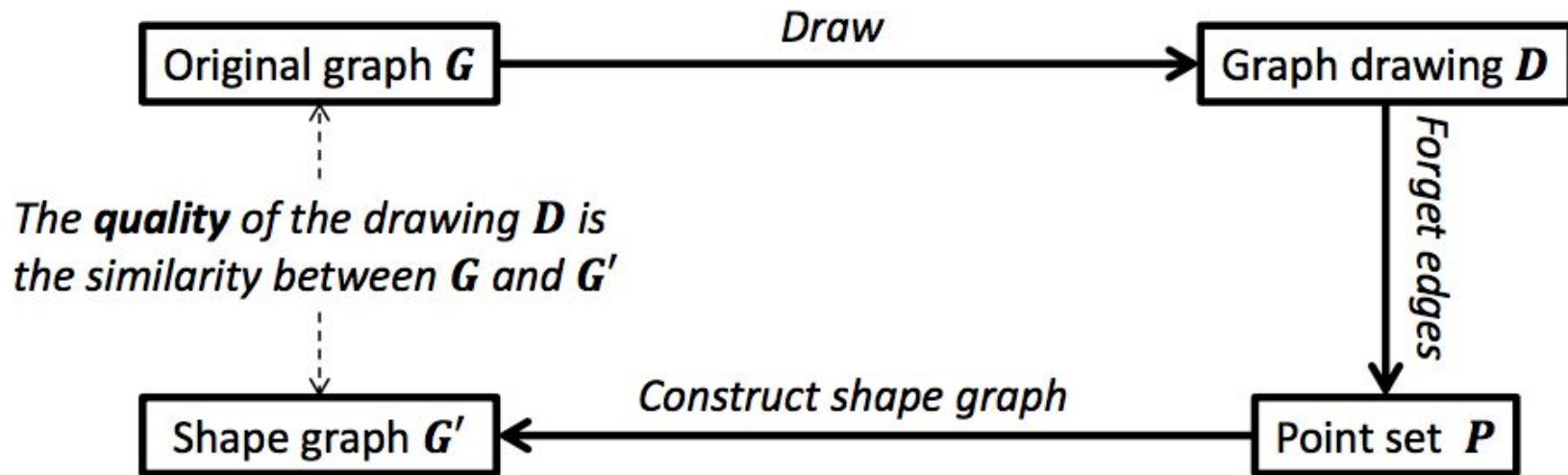
# Эстетические метрики

- Минимизировать пересечение ребер
- Максимизировать минимальный угол между инцидентными ребрами вершины
- Максимизировать угол между пересекающимися ребрами
- Унификация длин ребер
- Стресс
- Основанные на форме (Shape-based)

# Shape graphs

- Граф k-ближайших соседей (k-nearest neighbours)
- Триангуляции (triangulations)
- Граф Габриэля (Gabriel graph)
- Граф относительной близости (relative neighbourhood)
- Евклидово минимальное остовное дерево (Euclidean minimum spanning tree)

# Shape-based метрика



$$Q_{\mu, \eta}(D) = \eta(G, \mu(P))$$

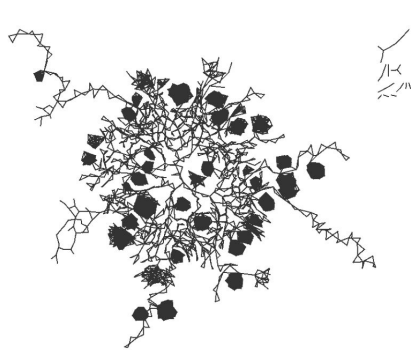


## Средняя схожесть Жаккара (Mean Jaccard similarity)

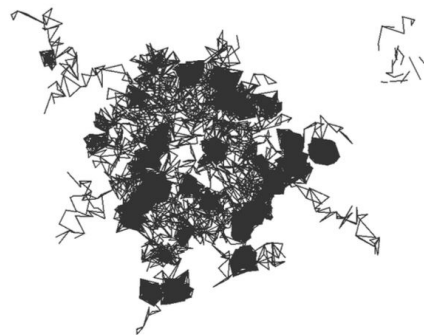
$$MJS(G_1, G_2) = \frac{1}{|V|} \sum_{u \in V} \frac{|N_1(u) \cap N_2(u)|}{|N_1(u) \cup N_2(u)|},$$

$N_i(u)$  - множество соседей  $u$  в графе  $G_i$  для  $i = 1, 2$

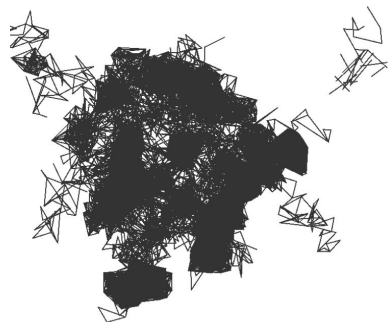
# Эксперименты



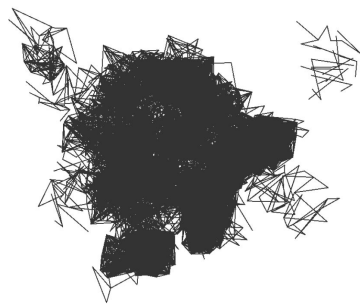
Изначальная раскладка



5 шагов деформации



10 шагов деформации



15 шагов деформации

Специально портим раскладку графа - в случайном направлении разбрасываем вершины на случайном расстоянии  $[0, \delta \cdot w]$ ,  $w$  - ширина экрана - и смотрим на то, как изменяется метрика качества

# Эксперименты

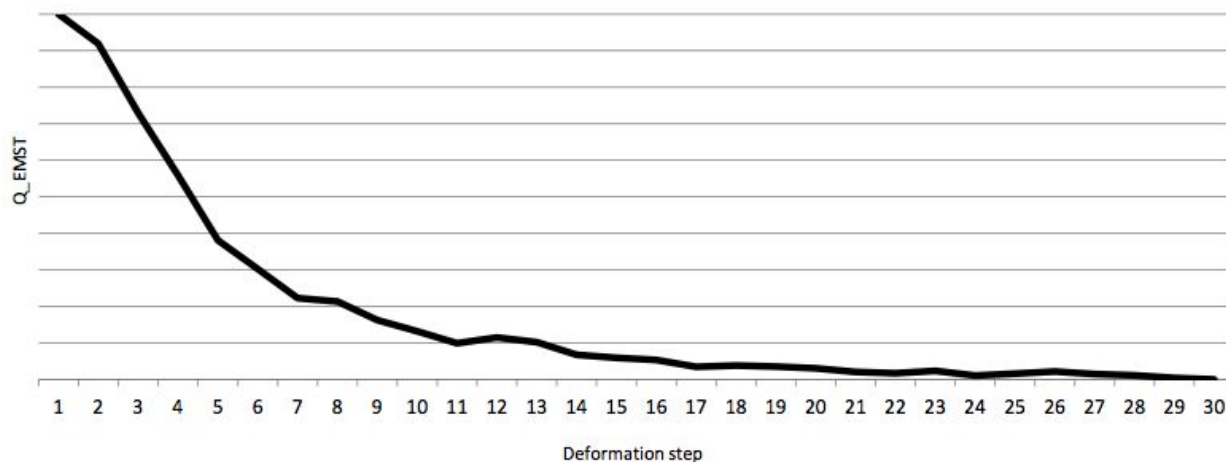


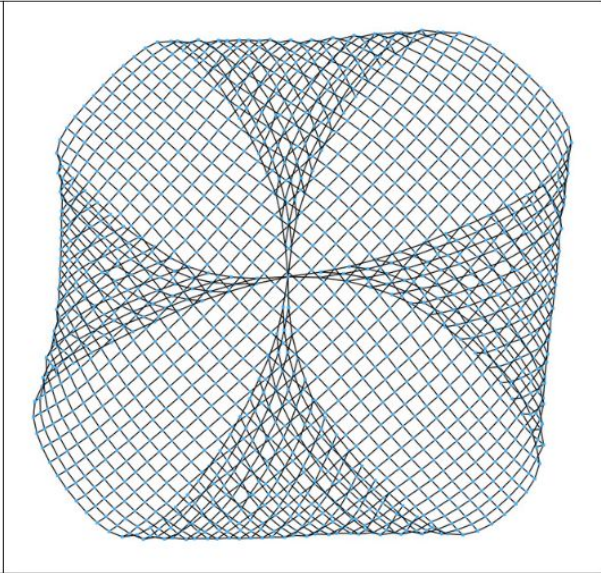
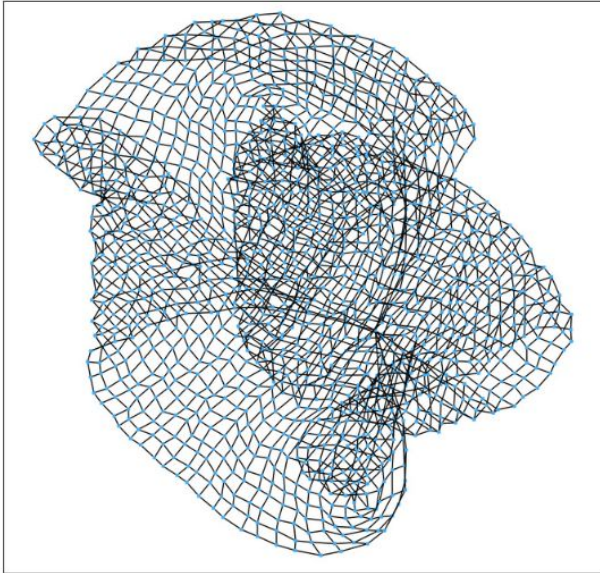
Figure 12:  $Q_{EMST}$  values as the drawing *stringyBlobsOrganic* is deformed.

Чем больше деформируем раскладку, тем ниже метрика

# Эксперименты

Please indicate your preference

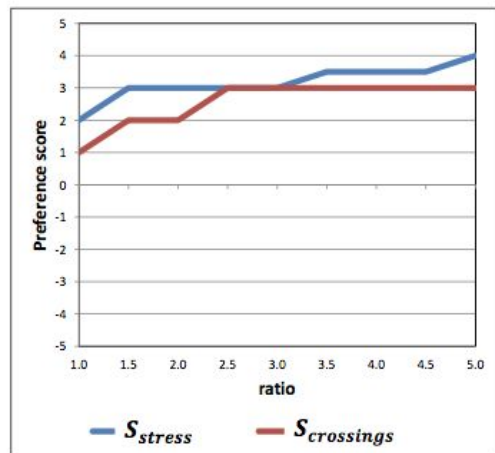
By dragging the slider to the left for the if you prefer the left drawing, and to the right if you prefer the right drawing.



5 4 3 2 1 0 1 2 3 4 5

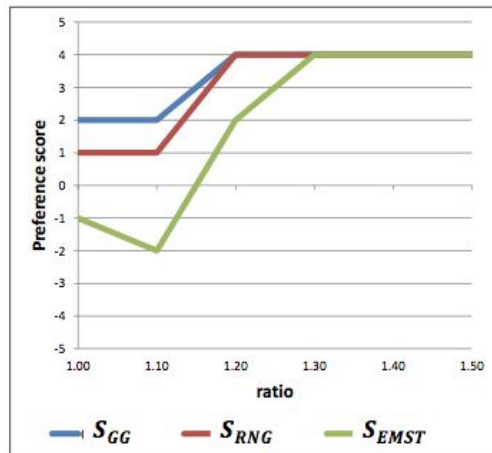
Next

# Эксперименты



(a)

Пользователи предпочитают графы с меньшим числом пересечений и стресса



(b)

Пользователи предпочитают графы с большим значением новой метрики, причем новая метрика лучше, чем стресс и пересечения соотносится с предпочтениями пользователя

$$Q\text{-ratio} = \frac{\max(Q_{\mu}(D_{left}), Q_{\mu}(D_{right}))}{\min(Q_{\mu}(D_{left}), Q_{\mu}(D_{right}))}.$$

Каждый элемент выборки может иметь значение предпочтения  $0 \leq x \leq 5$ ,  $S(I) = x$ , если элемент имеет большее значение метрики качества, иначе  $S(I) = -x$ , далее для каждого элемента брали медиану этого значения

# Эксперименты

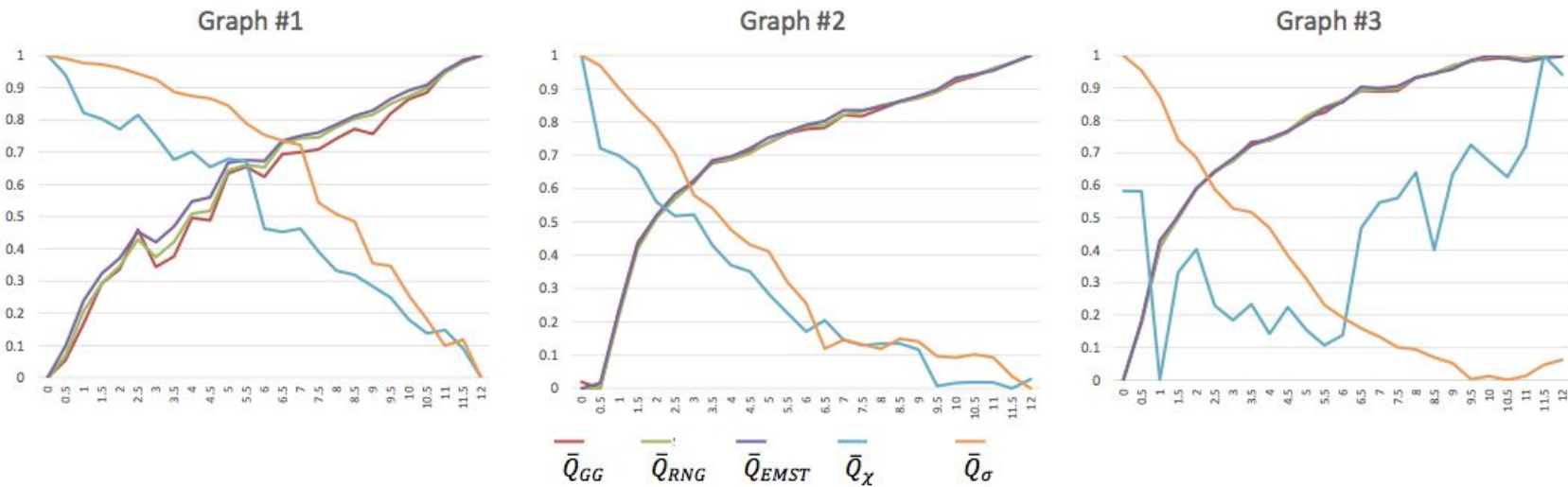


Figure 15: *Metrics against untangling.*

# Эстетические метрики - в одну?

Отсутствие метрики, измеряющей качество раскладки в целом => оцениваем раскладку, основываясь на

- личном суждении
- пользовательских исследованиях

Личное суждение - слишком субъективно

Пользовательские исследования - дорого

# Эстетические метрики - в одну?

1. Минимизировать количество пересечений (*cross#*)
2. Максимизировать угол пересечения (*crossRes*)
3. Максимизировать угловое разрешение вершины (*angularRes*)
4. Унифицировать длины ребер (*uniEdge*)

$$O = -Z_{cross\#} + Z_{crossRes} + Z_{angularRes} - Z_{uniEdge}$$

$$Z_{cross\#} = \frac{x - Mean}{StDev}$$



# Эксперимент

Пользователь должен выполнить какое-то задание на графах, в данном эксперименте - найти кратчайший путь между двумя вершинами, при этом эксперимент ставится так, что этот путь единственный и одна из вершин уже указана

Измеряем *время*, затраченное на выполнение задания, *усилия*, а также *точность*. Исследуем *чувствительность* (соотношение заранее известного качества раскладок с качеством, измеренным новой метрикой) и *прогнозируемость* (исследуем зависимость между метриками эффективности понимания графа человеком и новой метрикой)

# Чувствительность

**Чувствительность** - сигнализирует о разнице, если есть изменения в качестве (соответствует ли качество, измеренное метрикой, действительному качеству)

## Эксперимент:

- генерируется множество случайных графов со схожей структурой
- к каждому применялся force-directed метод раскладки
- взяли раскладки на 3000, 6000, 9000 и 12000 шагах алгоритма
- выясняется, соотносится ли измеренное новое метрикой общее качество раскладок с заранее известными уровнями качества

# Чувствительность

$$E = \frac{z_A - z_T - z_{ME}}{\sqrt{3}}$$

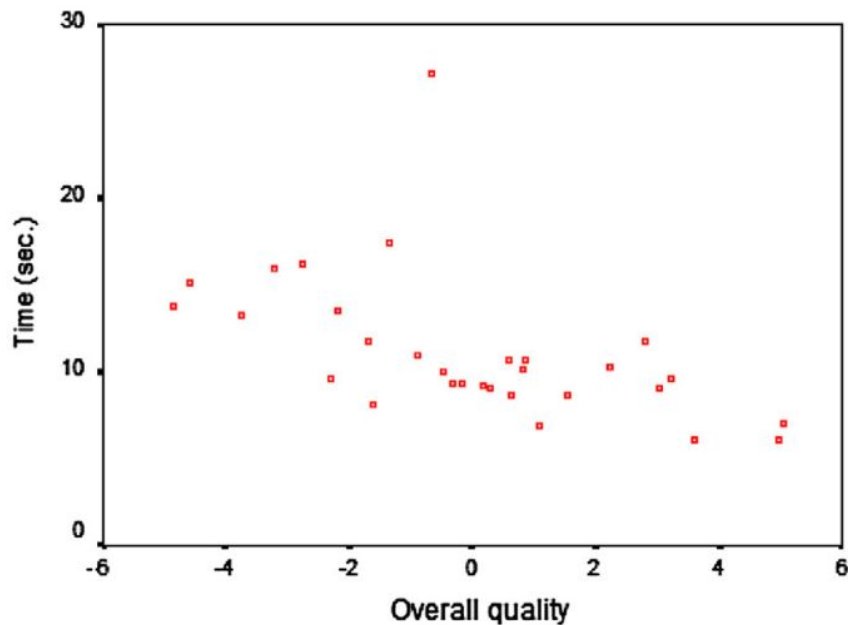
**Table 1**  
Mean values of dependent variables.

Variable	C3	C6	C9	C12
Time (s)	9.91	9.51	7.19	7.11
Effort	3.60	3.26	3.27	3.09
Accuracy	0.69	0.75	0.76	0.76
Efficiency	-0.74	-0.22	0.28	0.48
Overall quality	-2.14	0.04	1.02	1.08

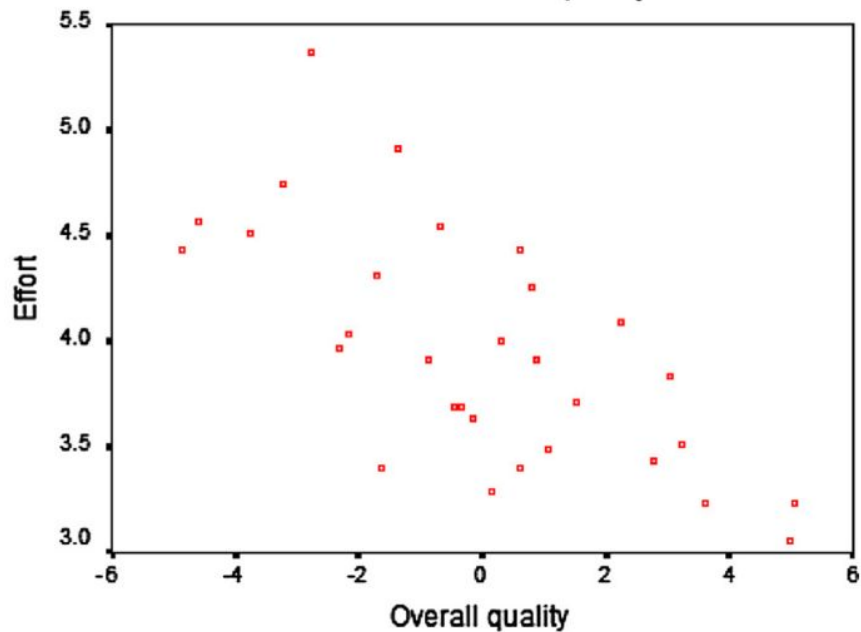
Можно видеть, что испытуемые обычно проводили меньше времени, прикладывали меньше усилий и были более точными, в то время как качество рисования улучшилось с c3 до c12.

# Прогнозируемость

Time vs. Overall quality

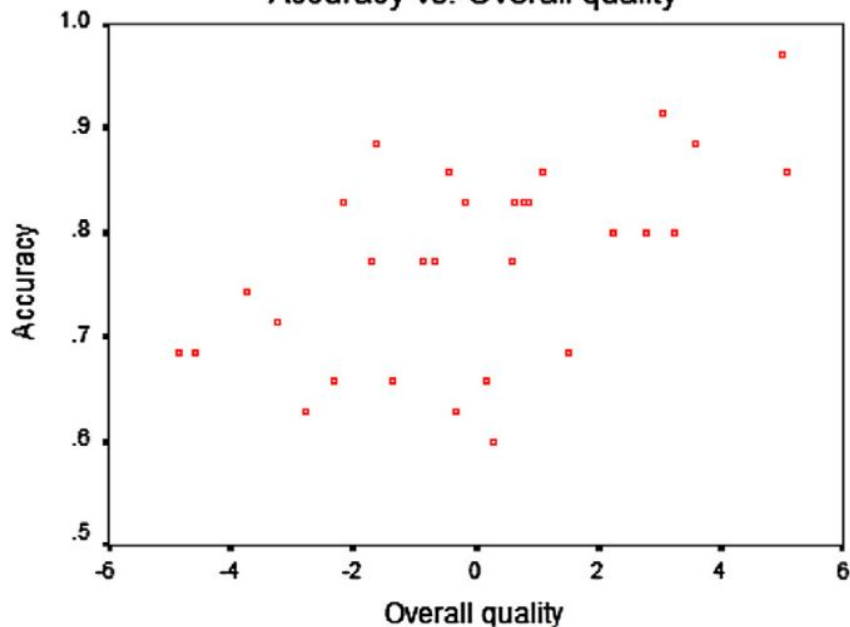


Effort vs. Overall quality

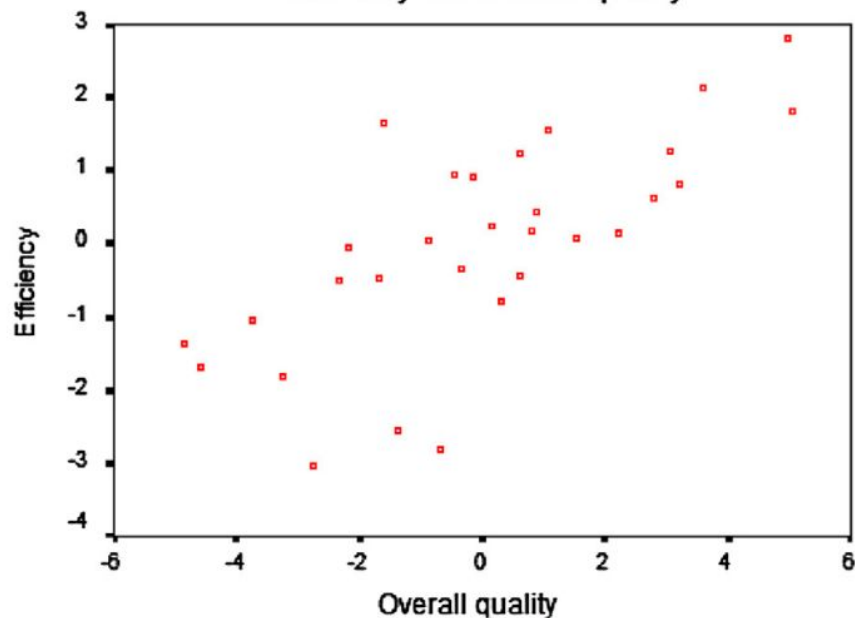


# Прогнозируемость

Accuracy vs. Overall quality



Efficiency vs. Overall quality



# Мотивация

- множество алгоритмов раскладки
- множество метрик для их оценки (причем нет общепринятого мнения, какие предпочтительнее)
- большой размер графов

=> нужно что-то, что быстрее, чем прямой перебор алгоритмов раскладки и подсчет для них всех метрик

# Graphlet kernel

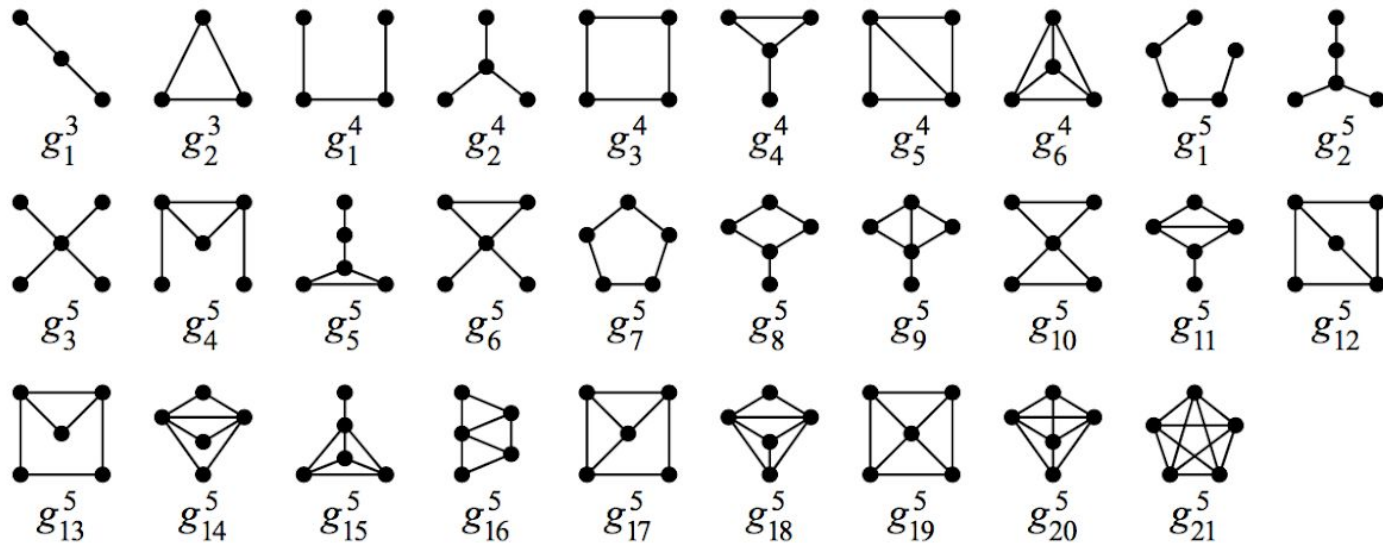


Fig. 2. All connected graphlets of 3, 4, or 5 vertices.

# Graphlet kernel

## 1. Выбрать алгоритм для выборки graphlet

- a. Random Vertex
- b. Random Walk

## 2. Отмасштабировать вектора частоты graphletов

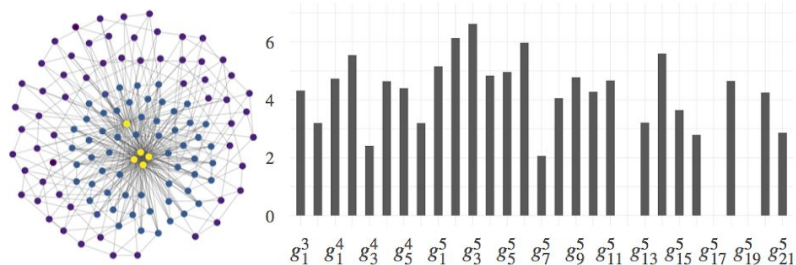
- a. линейное масштабирование  $x_i = \frac{w_i}{\sum w_i}$
- b. логарифмическое  $x_i = \log \left( \frac{w_i + w_b}{\sum (w_i + w_b)} \right)$

## 3. Определить скалярное произведение

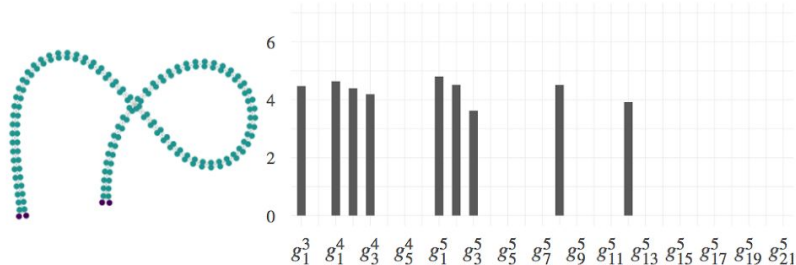
- a. косинусный коэффициент  $\langle \mathbf{x}, \mathbf{x}' \rangle = \frac{\mathbf{x} \cdot \mathbf{x}'^T}{\|\mathbf{x}\| \|\mathbf{x}'\|}$
- b. Gaussian radial basis function kernel  $\langle \mathbf{x}, \mathbf{x}' \rangle = \exp \left( -\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2} \right)$
- c. Laplacian kernel  $\langle \mathbf{x}, \mathbf{x}' \rangle = \exp \left( -\frac{\|\mathbf{x} - \mathbf{x}'\|_1}{\sigma} \right)$



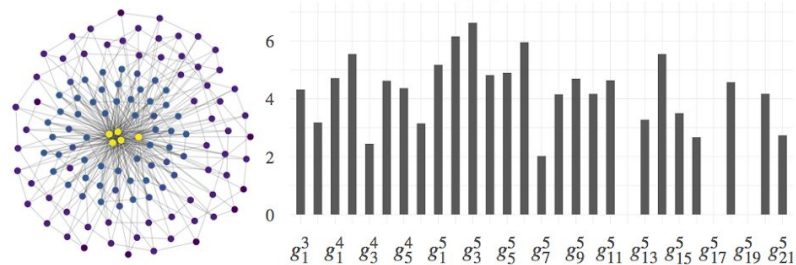
# What Would a Graph Look Like in This Layout



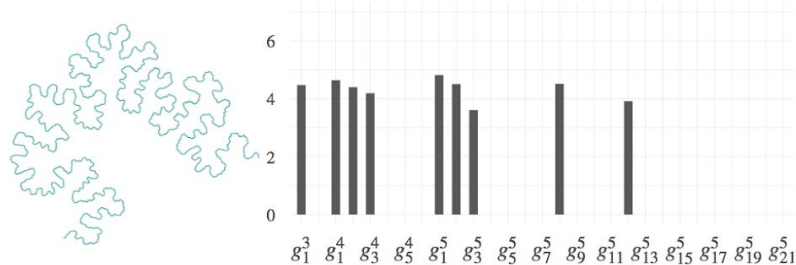
(a)  $G_{883}$   $|V| = 122$   $|E| = 472$



(b)  $G_{1788}$   $|V| = 158$   $|E| = 312$



(c)  $G_{943}$   $|V| = 124$   $|E| = 462$



(d)  $G_{7208}$   $|V| = 19,998$   $|E| = 39,992$

Не всегда схожесть частот graphletов говорит о топологической схожести графов => добавили ограничение на отношение количества вершин

# Метрики

$$m_c = \begin{cases} 1 - \frac{c}{c_{\max}}, & \text{if } c_{\max} > 0 \\ 1, & \text{otherwise} \end{cases} \quad c_{\max} = \frac{|E|(|E| - 1)}{2} - \frac{1}{2} \sum_{v \in V} (\deg(v)(\deg(v) - 1))$$

$$m_a = 1 - \frac{1}{|V|} \sum_{v \in V} \left| \frac{\theta(v) - \theta_{\min}(v)}{\theta(v)} \right|, \quad \theta(v) = \frac{360^\circ}{\deg(v)}$$

# Метрики

$$m_l = \frac{l_{cv}}{\sqrt{|E| - 1}}, \quad l_{cv} = \frac{l_\sigma}{l_\mu} = \sqrt{\frac{\sum_{e \in E} (l_e - l_\mu)^2}{|E| \cdot l_\mu^2}}$$

$$m_s = \text{MJS}(G_{\text{input}}, G_S), \quad \text{MJS}(G_1, G_2) = \frac{1}{|V|} \sum_{v \in V} \frac{|N_1(v) \cap N_2(v)|}{|N_1(v) \cup N_2(v)|}$$

# What Would a Graph Look Like in This Layout

1. Обучаем регрессионную модель на входных данных
2. Подсчитываем похожесть для входного графа и уже существующих графов (для которых уже рассчитаны раскладки)
3. Вычисляем значение метрики, используя обученную модель

# Оценки точности

$$\text{RMSE}(\mathcal{Y}, \tilde{\mathcal{Y}}) = \sqrt{\frac{1}{n} \sum_i (y_i - \tilde{y}_i)^2}$$

- среднеквадратичная ошибка

$$R^2(\mathcal{Y}, \tilde{\mathcal{Y}}) = 1 - \sum_i (y_i - \tilde{y}_i)^2 / \sum_i (y_i - y_\mu)^2$$

коэффициент детерминации

# Результаты

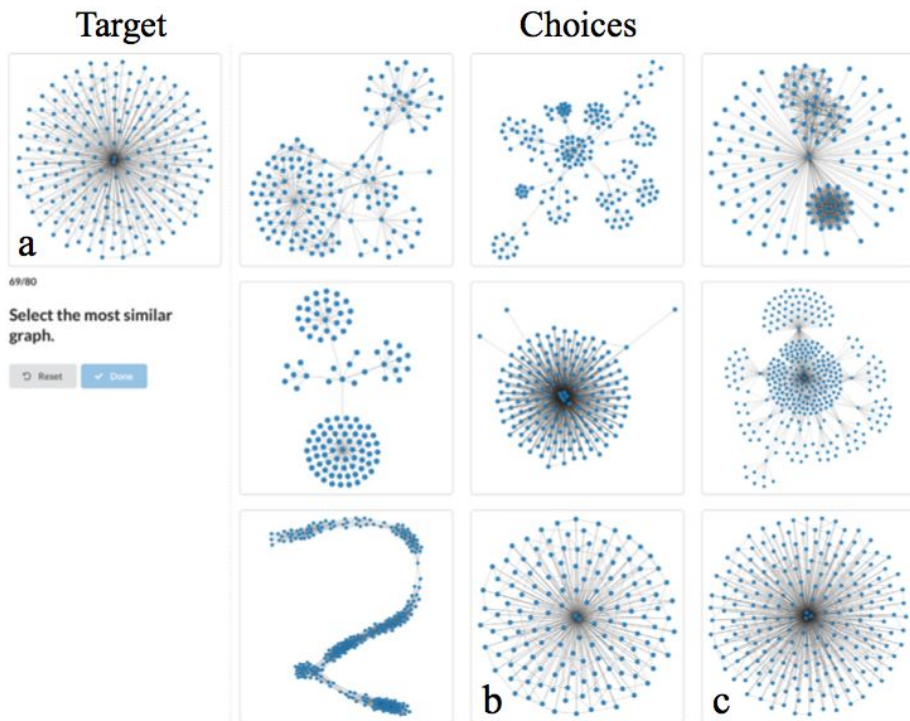
Table 1. Estimation accuracy of the two most accurate kernels and the state-of-the-art kernels. We report Root-Mean-Square Error (RMSE) and the coefficient of determination ( $R^2$ ) of estimation of four aesthetic metrics on eight layout methods.

Kernel		sfdp		FM <sup>3</sup>		FR		KK		Spectral		HDE		Treemap		Gosper	
		RMSE	$R^2$	RMSE	$R^2$	RMSE	$R^2$	RMSE	$R^2$	RMSE	$R^2$	RMSE	$R^2$	RMSE	$R^2$	RMSE	$R^2$
Rank 1 RW-LOG- LAPLACIAN	$m_c$	.0175	.9043	.0468	.7319	.0257	.8480	.0346	.8223	.1120	.6947	.0903	.8130	.0836	.6399	.0857	.6199
	$m_a$	.1011	.8965	.1041	.8919	.0982	.9004	.1024	.8876	.1153	.8793	.1152	.8666	.1053	.8552	.1071	.8580
	$m_l$	.0055	.9021	.0048	.8531	.0055	.9028	.0105	.4549	.0505	.6203	.0155	.5961	.0047	.8666	.0066	.8444
	$m_s$	.0514	.9060	.0474	.9325	.0417	.8533	.0485	.9084	.0534	.9031	.0486	.8942	.0112	.8429	.0323	.7495
Rank 2 RW-LOG- RBF	$m_c$	.0176	.9036	.0446	.7568	.0279	.8218	.0350	.8182	.1138	.6845	.0917	.8072	.0841	.6356	.0882	.5976
	$m_a$	.1070	.8840	.1102	.8788	.1023	.8920	.1061	.8793	.1193	.8706	.1202	.8546	.1101	.8416	.1125	.8434
	$m_l$	.0062	.8793	.0050	.8412	.0059	.8874	.0106	.4497	.0519	.5992	.0167	.5291	.0052	.8417	.0073	.8127
	$m_s$	.0556	.8900	.0542	.9116	.0459	.8227	.0547	.8833	.0576	.8875	.0537	.8708	.0116	.8299	.0323	.7491
Rank 11 RV-LIN- Cos [76]	$m_c$	.0387	.5312	.0771	.2716	.0577	.2364	.0783	.0916	.1533	.4280	.1770	.2827	.1324	.0978	.1336	.0763
	$m_a$	.2883	.1581	.2907	.1570	.2817	.1805	.2850	.1292	.3019	.1723	.2978	.1080	.2688	.0557	.2726	.0801
	$m_l$	.0168	.0972	.0121	.0561	.0169	.0895	.0138	.0609	.0812	.0200	.0239	.0403	.0116	.2026	.0156	.1378
	$m_s$	.1721	-.0552	.1904	-.0890	.0984	.1850	.1653	-.0656	.1777	-.0729	.1538	-.0606	.0246	.2373	.0628	.0521
Rank 12 DGK [91]	$m_c$	.0399	.5029	.0783	.2500	.0583	.2207	.0803	.0448	.1564	.4041	.1804	.2541	.1358	.0489	.1345	.0630
	$m_a$	.2891	.1536	.2924	.1467	.2837	.1690	.2862	.1217	.3052	.1537	.3003	.0930	.2716	.0357	.2754	.0612
	$m_l$	.0175	.0246	.0126	-.0134	.0177	.0047	.0140	.0294	.0811	.0203	.0243	.0018	.0128	.0029	.0185	-.3883
	$m_s$	.1756	-.0982	.1928	-.1171	.1077	.0236	.1676	-.0953	.1807	-.1094	.1550	-.0771	.0286	-.0846	.0682	-.1187

# What Would a Graph Look Like in This Layout

1. Подсчитываем похожесть для входного графа и уже существующих графов (для которых уже рассчитаны раскладки)
2. Убираем графы, которые не удовлетворяют набору ограничений
3. Берем  $k$  наиболее похожих графов
4. Показываем пользователю их раскладки для выбранного алгоритма

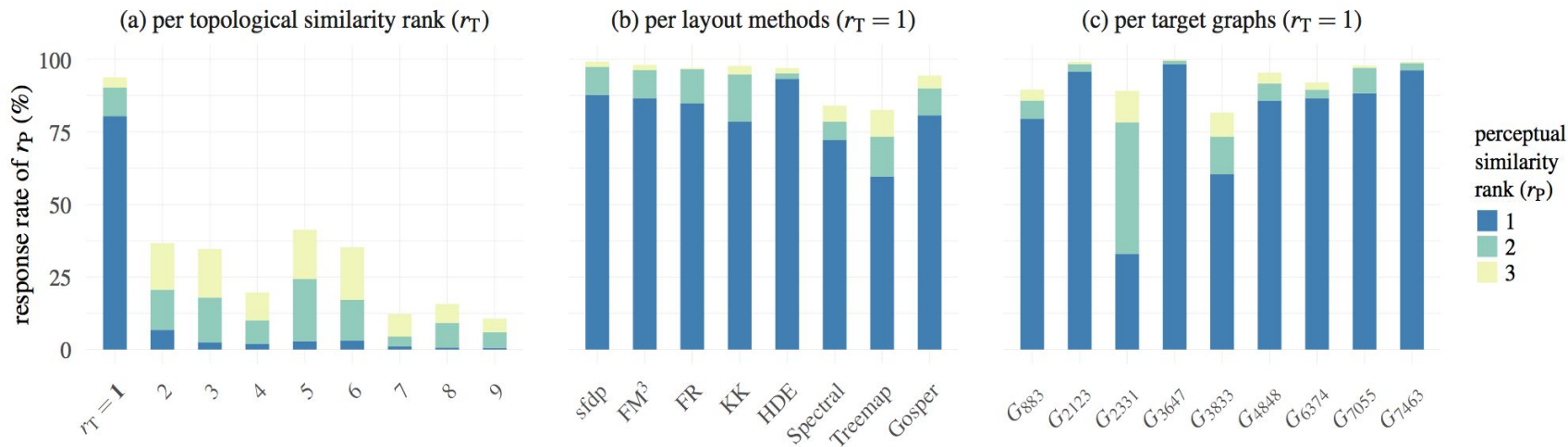
# Результаты



Пользователю предлагалось выбрать три наиболее похожих на исходных граф графа, для исходного графа и кандидатов использовался один и тот же метод раскладки, всего  $9 \times 8 = 72$  задания - 9 графов и 8 раскладок



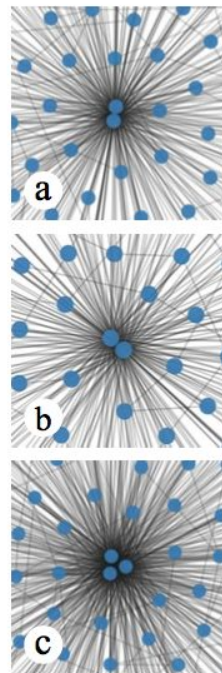
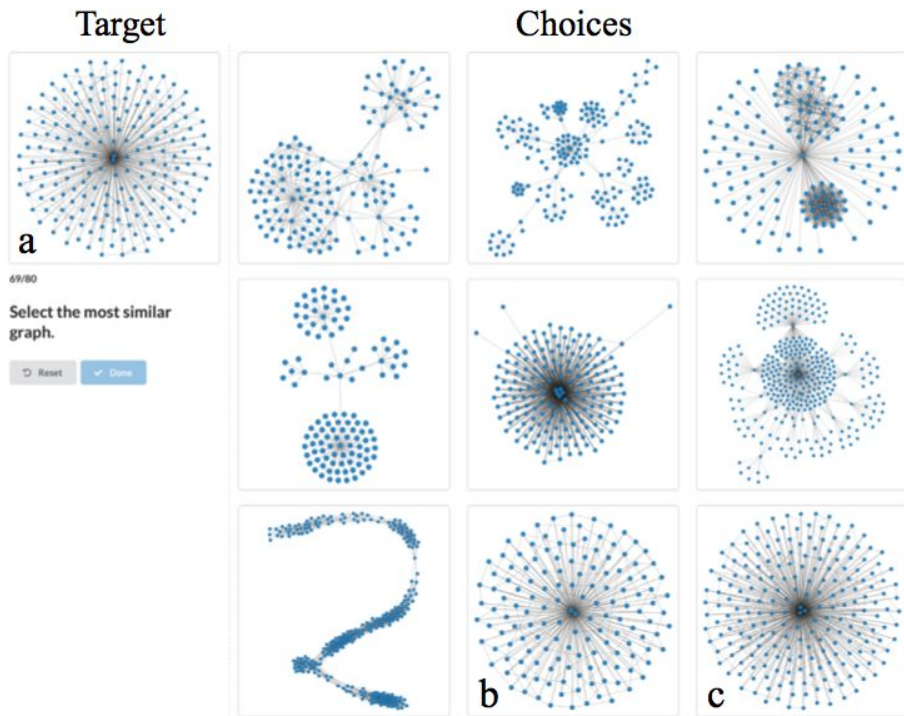
# Результаты



rt 1..9 - от самого “похожего” графа до самого непохожего

rp 1..3 - каким по счету пользователь выбрал этот граф как самый похожий

# Результаты



**По поводу графа 2331 с самым низким результатом:**  
участники выбирали вариант b, так как центр наиболее схож с графом на входе, а фреймворк рекомендовал c, из-за более схожей общей структуры

# Выводы

Разные методы выборки графа => разные восприятия пользователем => выбираем метод выборки исходя из того, какие “внешние качества” графа хотим сохранить

Метрика, основанная на форме графа, подходит для измерения качества раскладок и иногда более чувствительна к качеству, чем существующие

Метрики оценки качества раскладок графа важны, так как с помощью них можно оптимизировать алгоритмы и сравнивать результаты раскладок, объединяя некоторые метрики в одну, можно сравнивать качество раскладок в целом

Используя машинное обучение, можно ускорить процесс выбора раскладки, предсказывая внешний вид графа и прогнозируя метрики качества