



海洋时空挖掘与分析系统 (V2.0)

用户手册

2019 年 4 月

目录

1. 系统简介	3
1.1 系统主界面	3
1.2 数据库管理	3
1.2 时空信息提取	8
1.3 时空异常探测	8
1.4 时空聚类挖掘	8
1.5 时空聚类挖掘	9
1.6 时空数据查询	9
1.7 更换地图	12
1.8 帮助文档	12
2. 系统配置要求	12
3. 数据准备与导入（Neo4j 数据库）	12
3.1 数据要求与数据类型对照表	12
3.2 数据命名规则及关系文件格式	13
3.3 数据导入及过程-序列-状态模型构建	14
4. 功能说明与操作方法	16
附录 1. Neo4j 数据库要素属性表	17
附录 2. Neo4j 数据库常用操作与语句	17
附录 3. Neo4j 数据库配置与环境设置	17

1. 系统功能简介

1.1 系统主界面

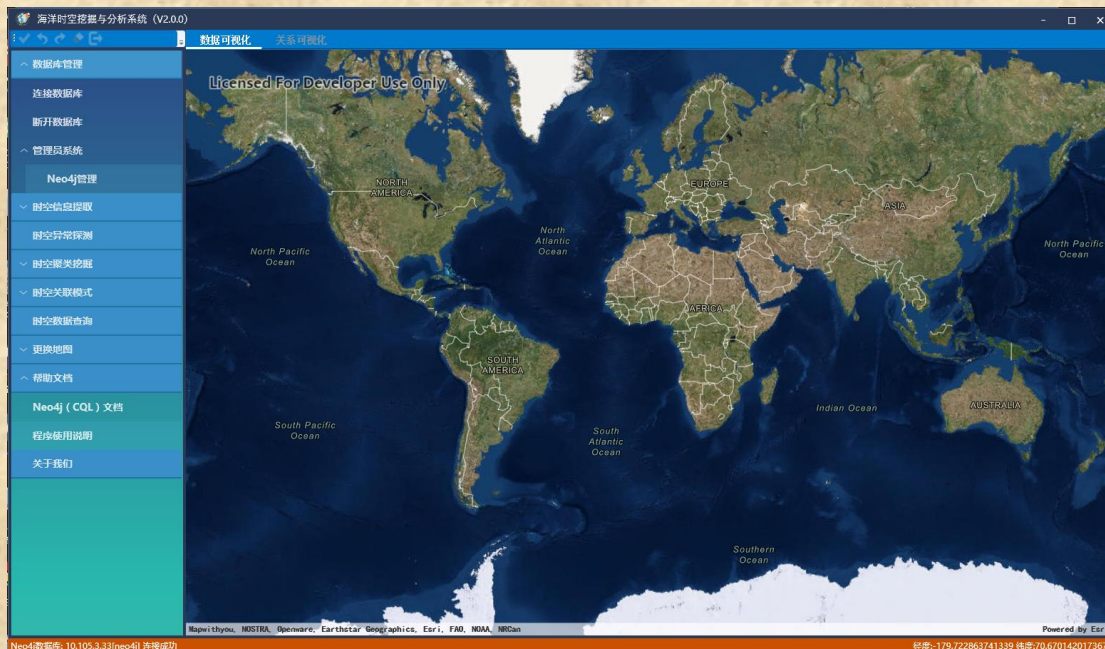


图 1-1 系统主界面

1.2 数据库管理

1) 连接数据库：

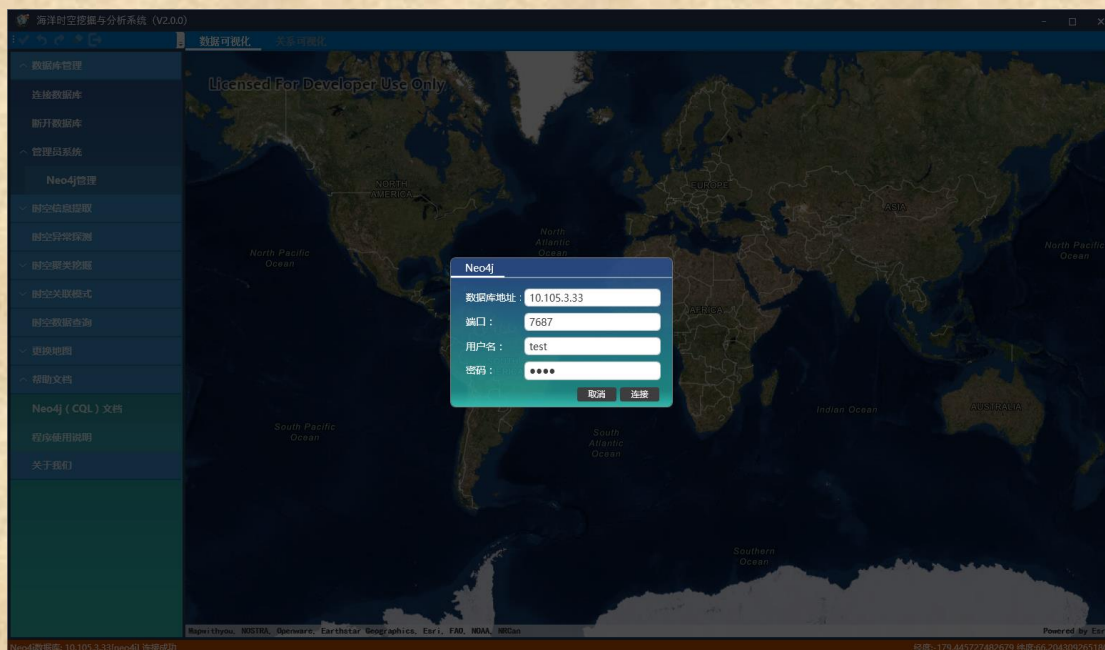


图 1-2 数据库登陆

- 2) 断开数据库：**该操作将断开所有数据库连接**
- 3) 管理员系统（不使用管理员用户登陆时仅部分功能可用）



图 1-3 矢量数据导入

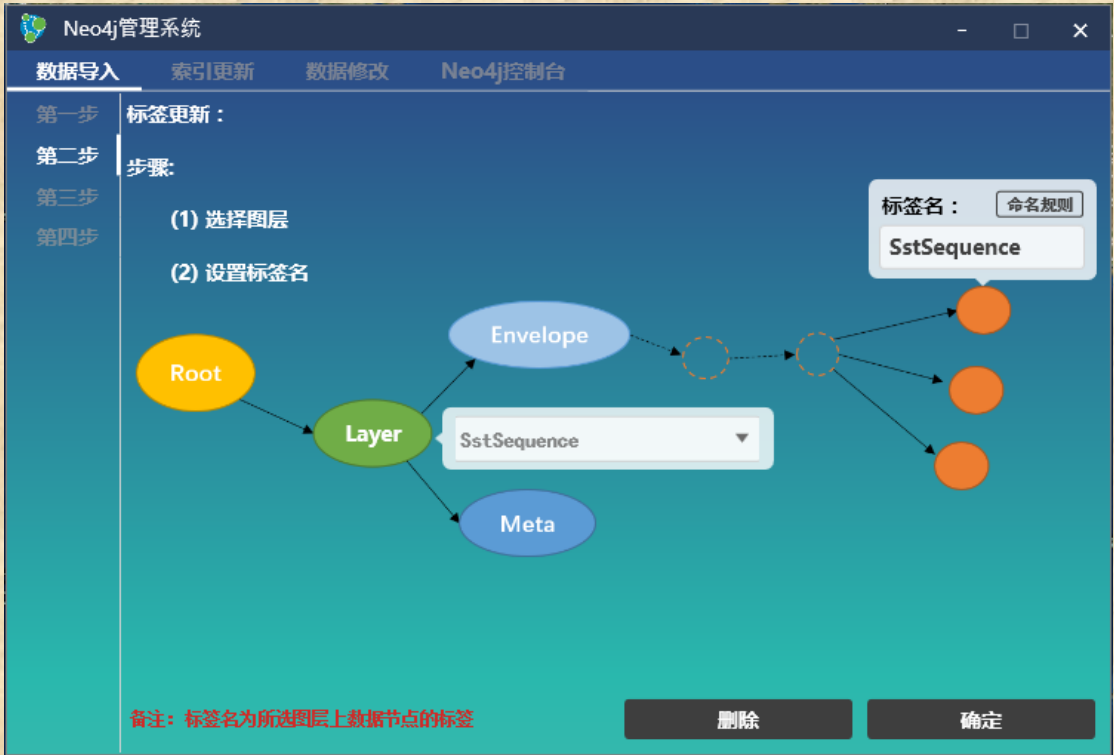


图 1-4 标签更新

【备注】：通过数据导入.Shp 文件后，节点无实际标签名，无法进行数据查询、关系创建、索引更新等。需要根据导入数据所在图层，为节点创建节点标签名（命名规则可参照 3.2）

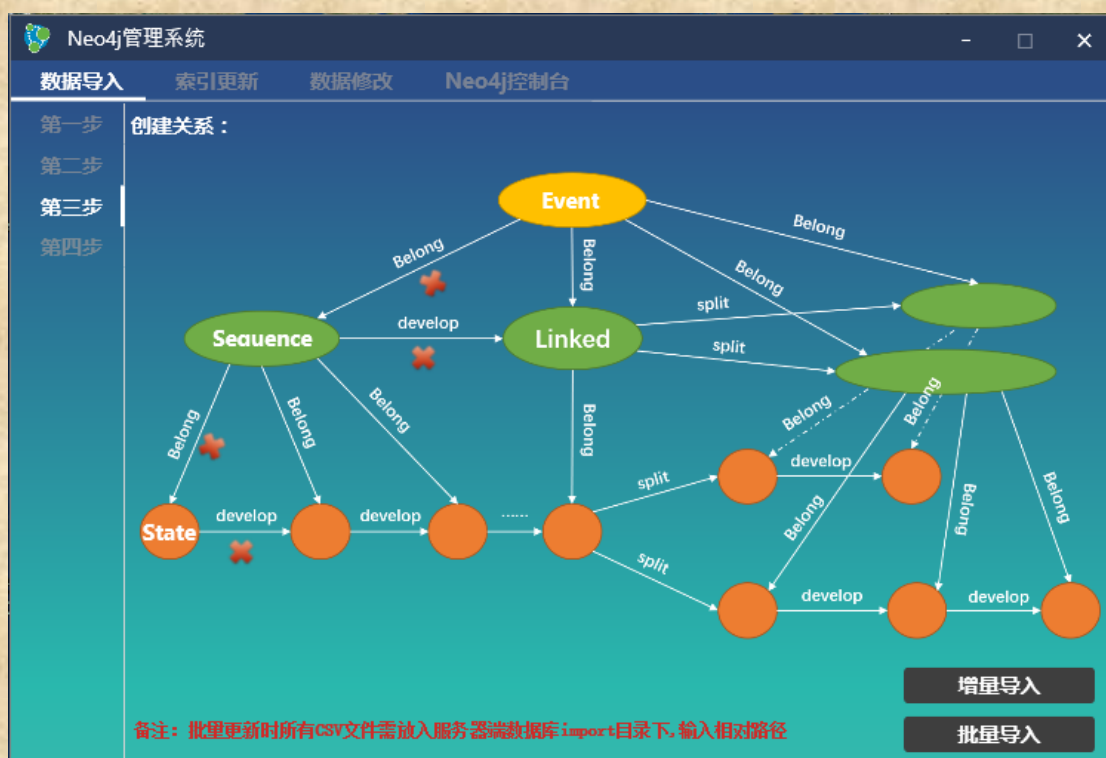


图 1-5 （a）关系创建

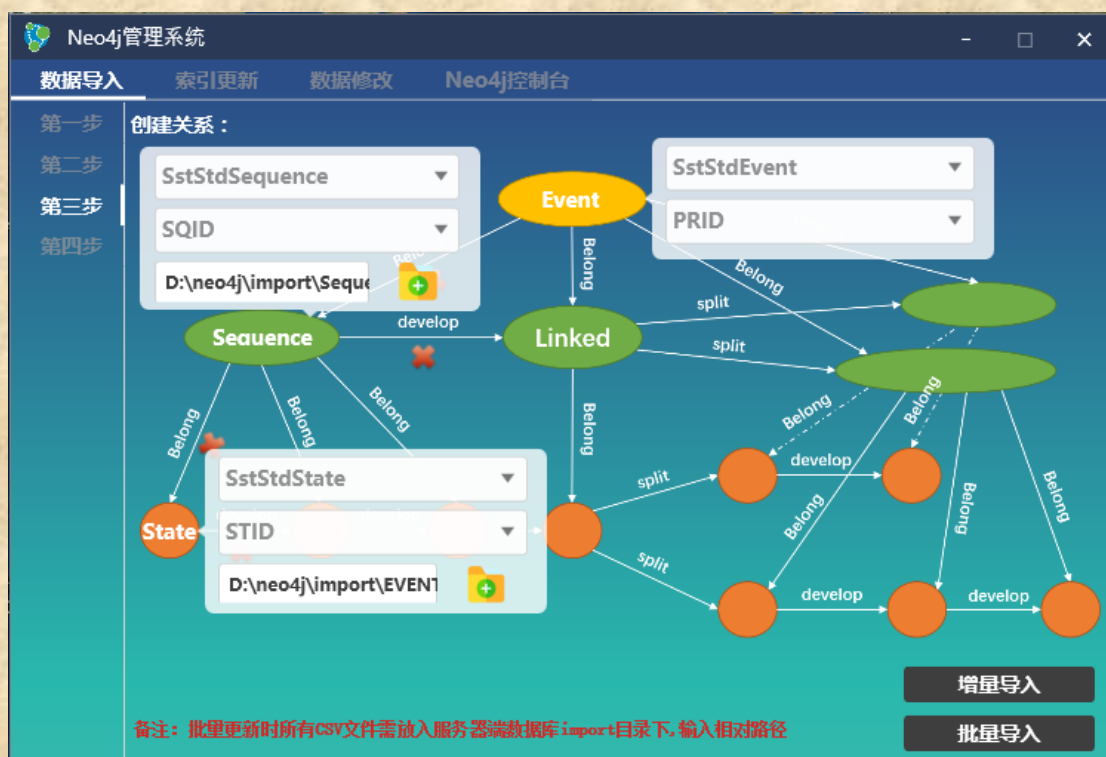


图 1-5 （b）关系创建

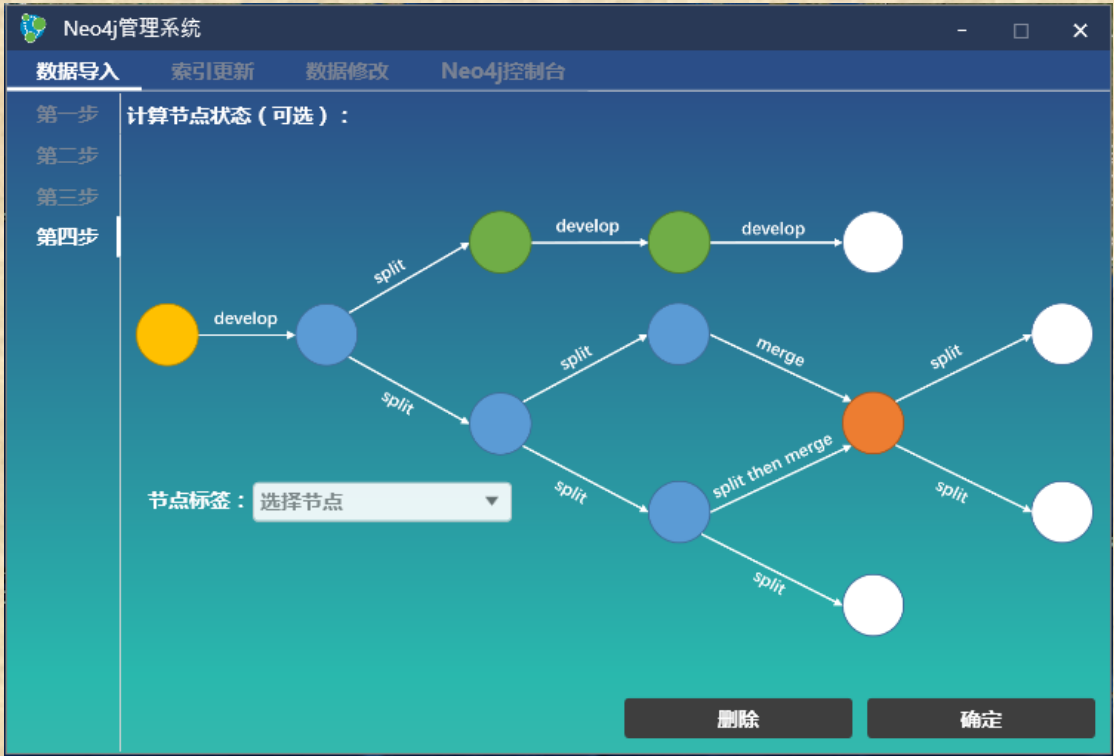


图 1-6 节点类型更新

【备注】：节点状态将根据节点出入度设置 TYPE 字段，具体规则参见《附录 1.常用操作与语句》中节点状态类型分类

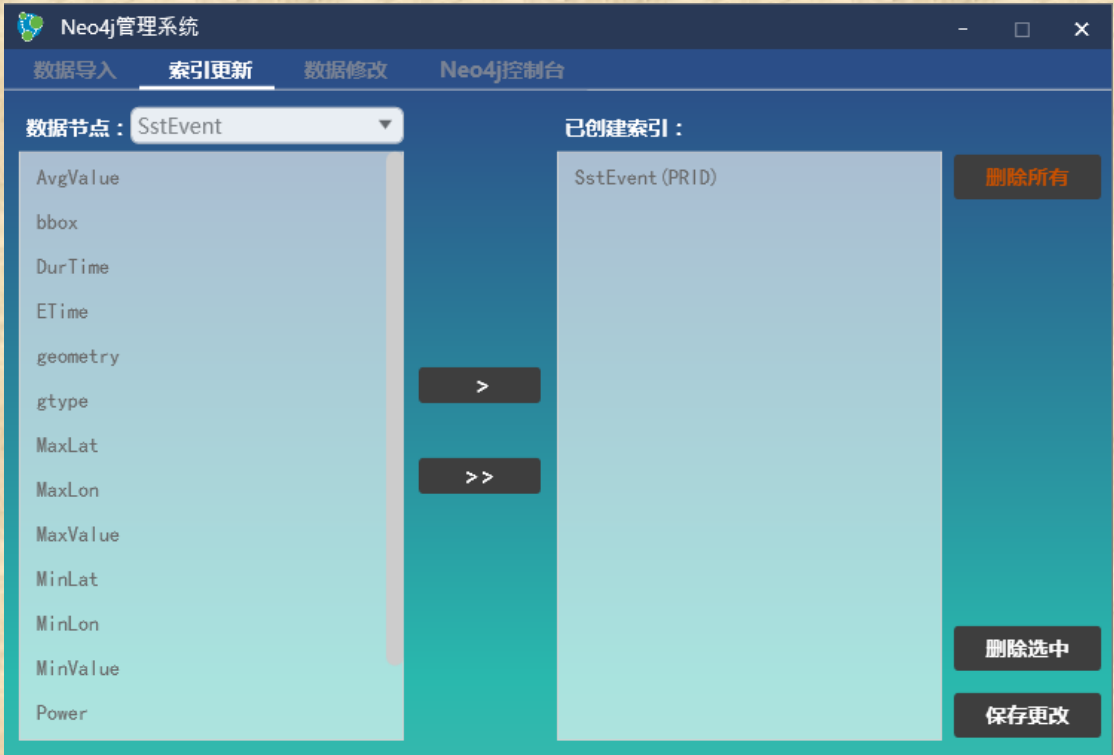


图 1-7 索引更新

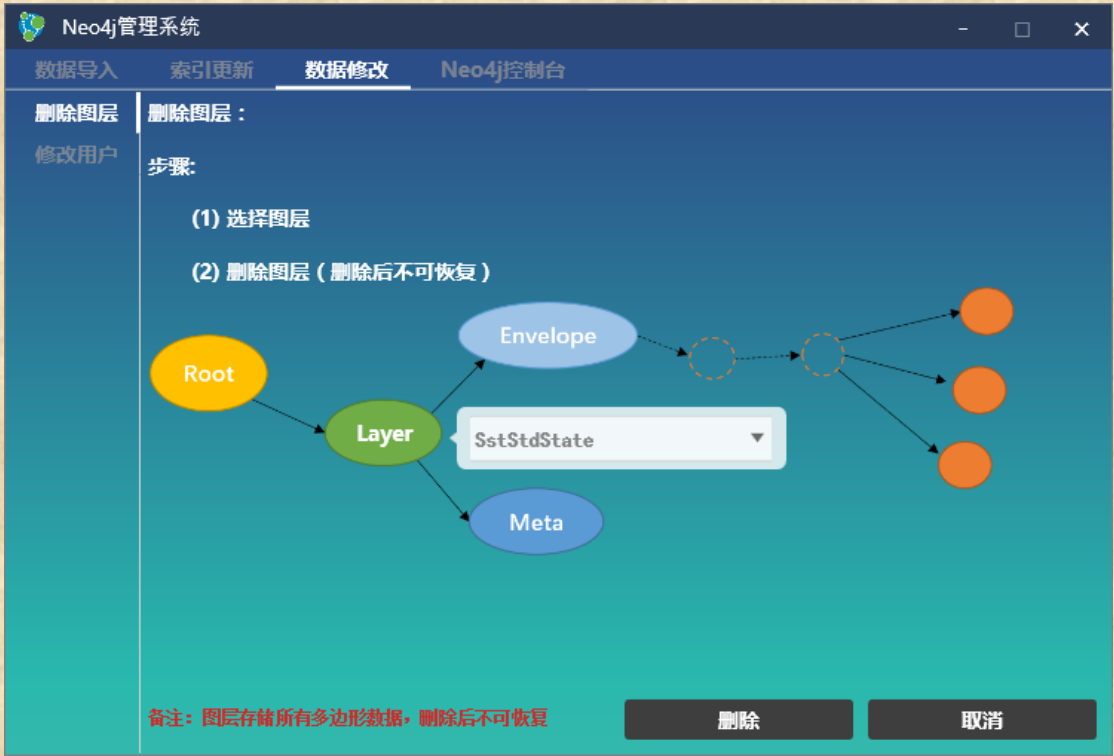


图 1-8 删除图层



图 1-9 用户修改

【备注】：只有具有 admin 角色的用户可以进行修改操作，程序运行查询等操作至少需要 reader 角色

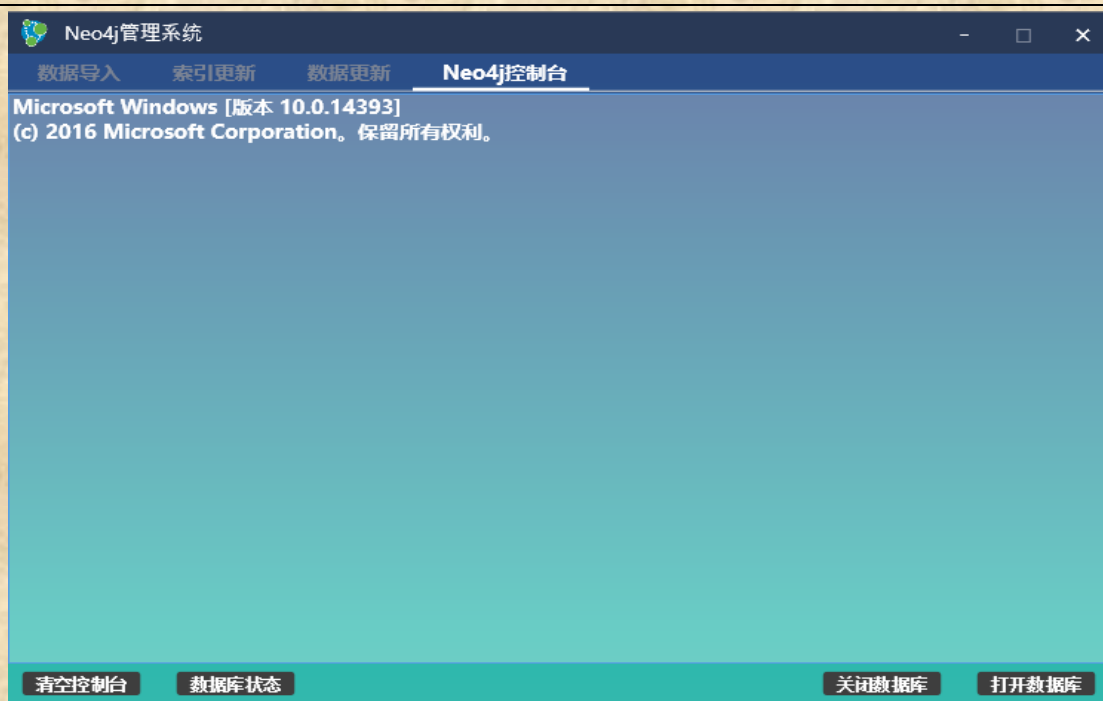


图 1-10 Neo4j 控制台（在 Neo4j 所在的服务器上可运行）

1.2 时空信息提取

1.3 时空异常探测

1.4 时空聚类挖掘

a) 面向栅格聚类

- i. 基础聚类
- ii. 空间聚类
 1. RoCMSAC
 2. DDBSC
- iii. 时间序列聚类
- iv. 时空聚类
 1. DcSTCA

2. ST-SNN

3. ST-DBSCAN

b) 面向对象聚类

c) 面向过程聚类

1.5时空聚类挖掘

a) 时空关联挖掘事物表构建

i. 对象事物表

ii. 事件事物表

b) 面向栅格关联模式挖掘

c) 面向对象关联模式挖掘

d) 面向过程关联模式挖掘

1.6时空数据查询

a)属性条件

b)时间条件

c)空间条件

图 1-10 时空查询



图 1-11 查询结果表

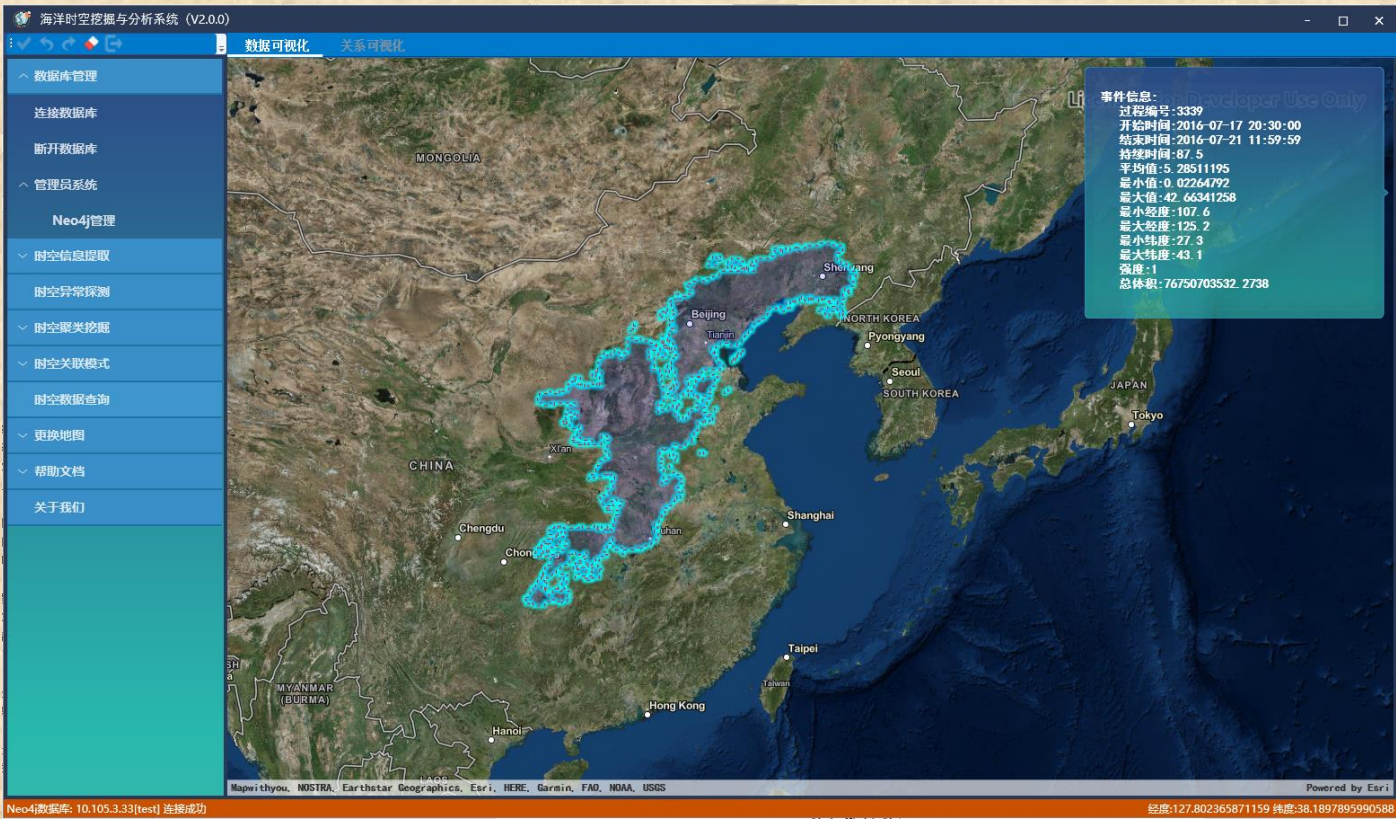


图 1-12 查询结果可视化

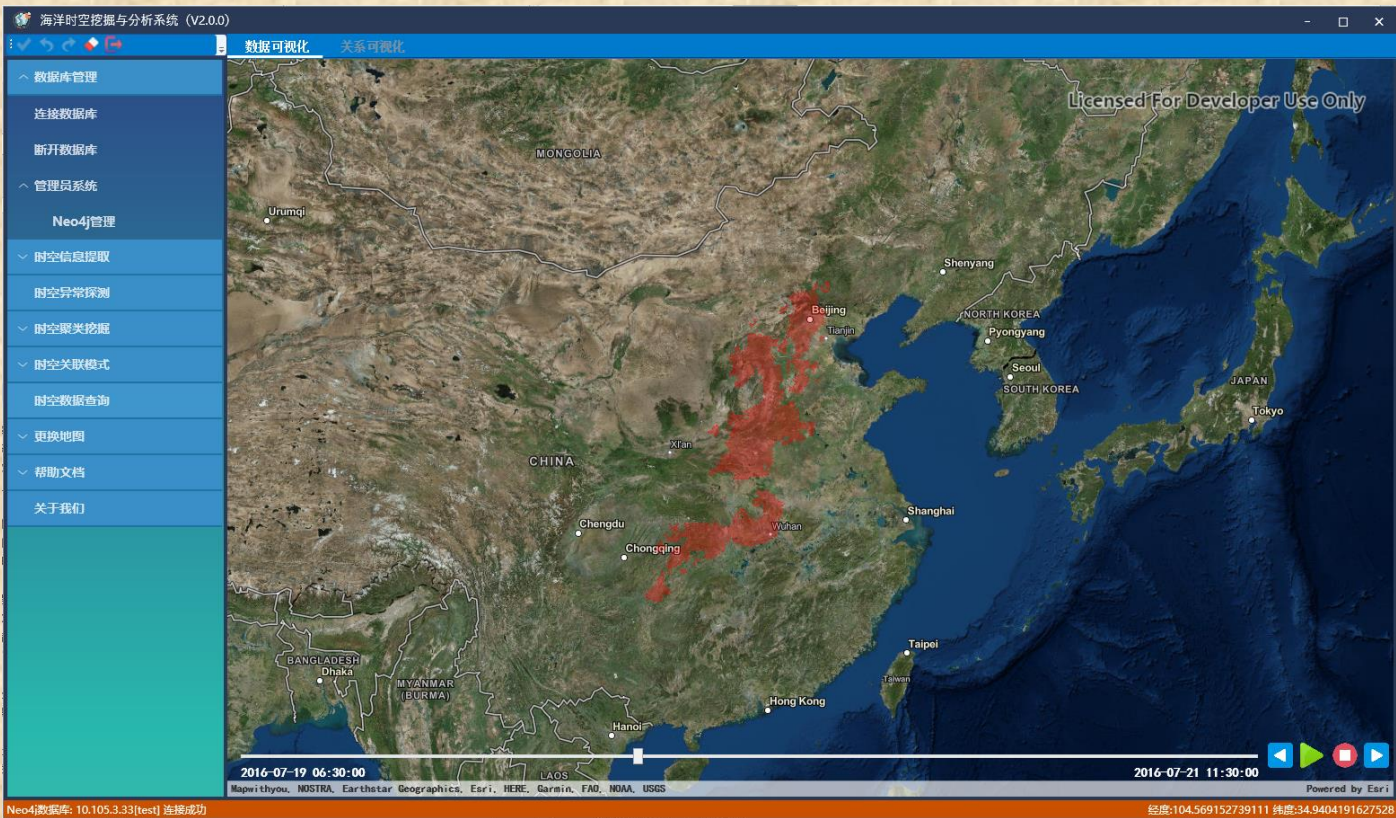


图 1-12 查询结果动画播放

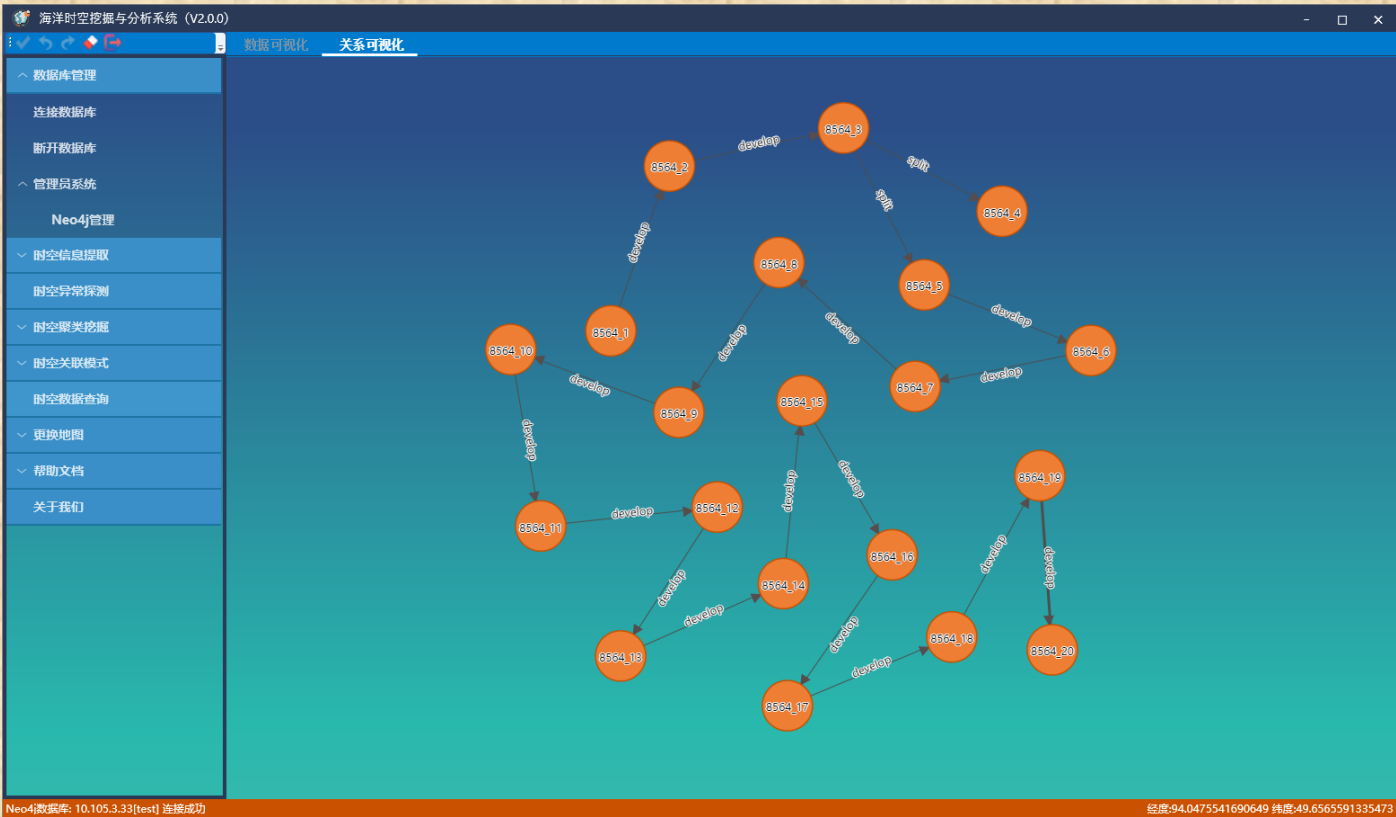


图 1-13 查询结果图结构

1.7 更换地图

1.8 帮助文档

2. 系统配置要求

2.1 硬件配置及软件环境

表 2-1.配置表

	最低配置	推荐配置
CPU	2.7 GHz Intel Core i5	Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz
内存	4GB DDR3	8GB DDR3
GPU	Intel Iris Graphics 6100	NVIDIA Quadro M6000 24 GB
硬盘	128g SSD/128g HDD	128g SSD/512g HDD
系统	Windows10 64 位 .NET Framework 4.6.1	Windows Server2016 64 位 .NET Framework 4.6.1
【备注】：数据导入模块需要配置 JAVA JRE 1.8;.NET 已测试 4.601055 版本		

3. 数据准备与导入（Neo4j 数据库）

3.1 数据要求与数据类型对照表

1) 必要属性表：

表 2-2. 要素必需字段

过程					
过程编号 PRID	开始时间 STime	结束时间 ETime	WKT
序列					
过程编号 PRID	序列编号 SQID	开始时间 STime	结束时间 ETime	WKT	...
状态					
过程编号 PRID	序列编号 SQID	状态编号 STID	发生时间 Time	WKT	...
【备注】：PRID、SQID、STD 为编号字段，包含 Time 为时间相关，WKT 为多边形自动生成（矢量文件本身隐含 WKT 数据）					

2) 数据类型、中英文对照表:

表 3-1.数据类型、中英文对照表

字段名	数据类型	中文对照	字段名	数据类型	中文对照
PRID	INT	过程编号	MaxPower	DOUBLE	最大强度
SQID	STRING	序列编号	SumPower	DOUBLE	总强度
STID	STRING	状态编号	Volume	DOUBLE	体积
STime	DATETIME	开始时间	TotalVolume	DOUBLE	总体积
ETime	DATETIME	结束时间	CoreLon	DOUBLE	质心经度
DurTime	DATETIME	持续时间	CoreLat	DOUBLE	质心纬度
Time	DATETIME	发生时间	Area	DOUBLE	面积
Theta	DOUBLE	方向	MaxArea	DOUBLE	最大面积
AvgValue	DOUBLE	平均值	SumArea	DOUBLE	总面积
MinValue	DOUBLE	最小值	ValueType	STRING	异常类型
MaxValue	DOUBLE	最大值	StateType	INT	状态类型
MinLon	DOUBLE	最小经度	ParentsID	STRING	父节点
MaxLon	DOUBLE	最大经度	ChildrenID	STRING	子节点
MinLat	DOUBLE	最小纬度	gtype	INT	几何类型
MaxLat	DOUBLE	最大纬度	geometry	STRING	WKT
Power	INT	强度	bbox	STRING	外接矩形
MinPower	DOUBLE	最小强度	SeqType	STRING	序列类型

3.2 数据命名规则及关系文件格式

表 3-2.要素命名规则

命名方式	过程：要素名 (首字母大写)+(Event)	序列：要素名 (首字母大写)+(Sequence)	状态：要素名 (首字母大写)+(State)
暴雨	RainStormEvent	RainStormSequence	RainStormState
.....
【备注】：要素与图层同名			

表 3-3. 关系文件(.CSV)格式说明

*关系	前状态编号:	后状态编号:	关系类型	*方向
状态-状态	FromID	ToID	StateAction	方向:→
序列-状态	FromID	ToID	Belong	方向:→
过程-序列	FromID	ToID	Belong	方向:→
【备注】：关系文件中若不包含<关系类型>字段，默认为 Belong；关系文件由前状态编号、后状态编号和关系类型三列构成；StateAction 由实际的状态决定（分裂、合并等）；*列仅用于辅助说明				

3.3 数据导入及过程-序列-状态模型构建

1) 数据准备

- 过程、序列、状态文件：*.Shapefile 文件（数据目录应完整包含 dbf、shx 等文件）
- 序列-序列关系、状态-状态关系：*.csv 文件
- 过程-序列、序列-状态：通过 PRID 和 SQID 关联
- 数据库：graph.db 文件夹

2) 数据导入及模型构成

a) 矢量数据导入

表 3-4. 矢量数据导入接口

数据库接口		
“spatial.importShapefile”	Imports the provided shape-file from URI to a layer of the same name, returns the count of data added	spatial.importShapefile (uri:: STRING?) :: (count::INTERGER?)
“spatial.importShapefileToLayer”	Imports the provided shape-file from URI to the given layer, returns the count of data added	spatial.importShapefileToLayer (layerName:: STRING?, uri:: STRING?) :: (count::INTERGER?)
程序接口		
BatchShapefileImporter.importFile	JAVA 批量 shp 文件导入	void importFile (path, /* String*/ shpArray, /* ArrayList<String>*/ Layer, /* String */ charset /* Charset */)
【备注】： BatchShapefileImporter: https://github.com/WuChengbin/Neo4jShapeBatchImporter Neo4jSpatial: https://github.com/neo4j-contrib/spatial		

矢量数据可通过以下两种方式导入：

- 1、增量导入：可在线导入，只需具有 admin 权限的用户均可导入数据（并行速度由执行程序所在

的计算机 CPU 线程数决定)；

- 2、批量导入：无需 admin 权限，需在数据库服务器端停止数据库后，由 Java 程序直接导入到数据库数据文件中。速度由数据库服务器性能决定（较快）

b) 关系数据导入：详见附录 2.常用语句

关系数据可通过以下两种方式导入：

- 1、增量导入：通过 Cypher 语句逐条执行创建关系操作（并行速度由执行程序所在的计算机 CPU 线程数决定）；
- 2、批量导入：通过 Load CSV 命令加载数据库所在服务器的本地文件，批量创建关系（较快）

c) 以上两步操作执行成功后，可得到如下模型：

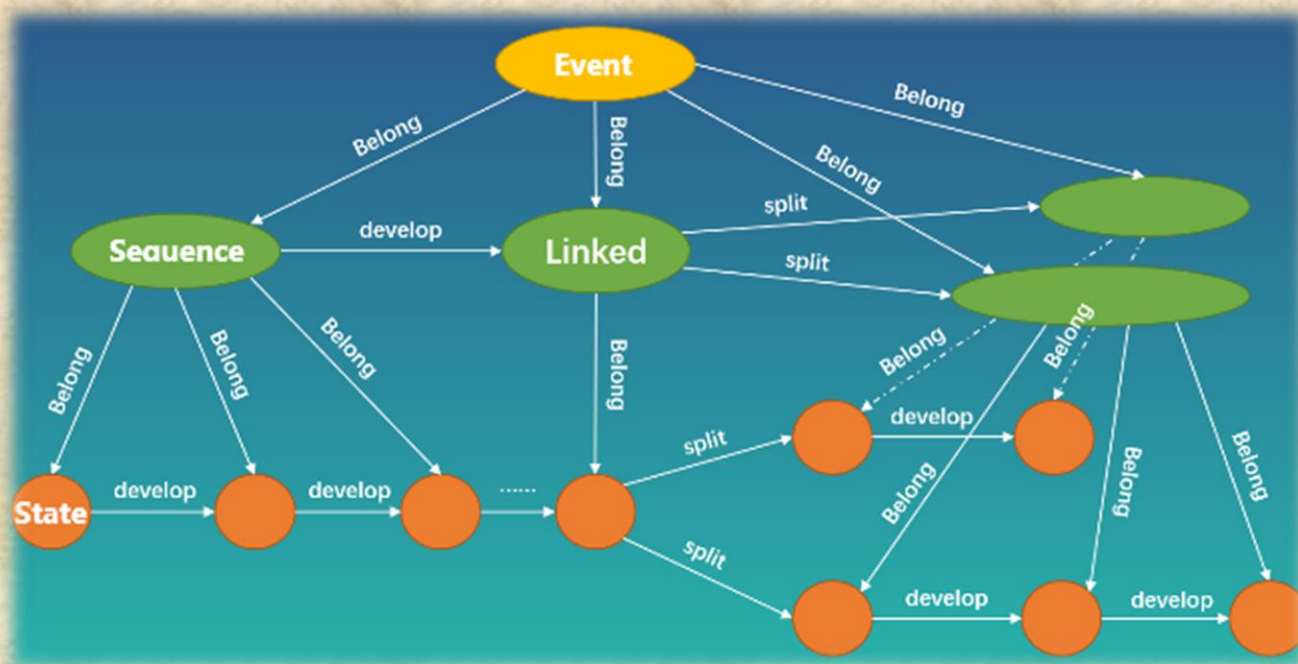


图 3-1 过程图模型

【备注】：模型理论思想和应用可参考以下文献：

【1】Xue, C.; Wu, C.; Liu, J.; Su, F. A Novel Process-Oriented Graph Storage for Dynamic Geographic Phenomena. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 100.

【2】李连伟, 伍程斌, 崔建勇, 刘敬一, & 薛存金. (2019). 基于图结构的暴雨事件组织方法研究. *系统工程理论与实践*, *39*(3), 805-816.

4. 功能说明与操作方法

4.1Neo4j 管理系统

1) 数据导入

a) 矢量数据导入



图 4-1 增量矢量数据导入

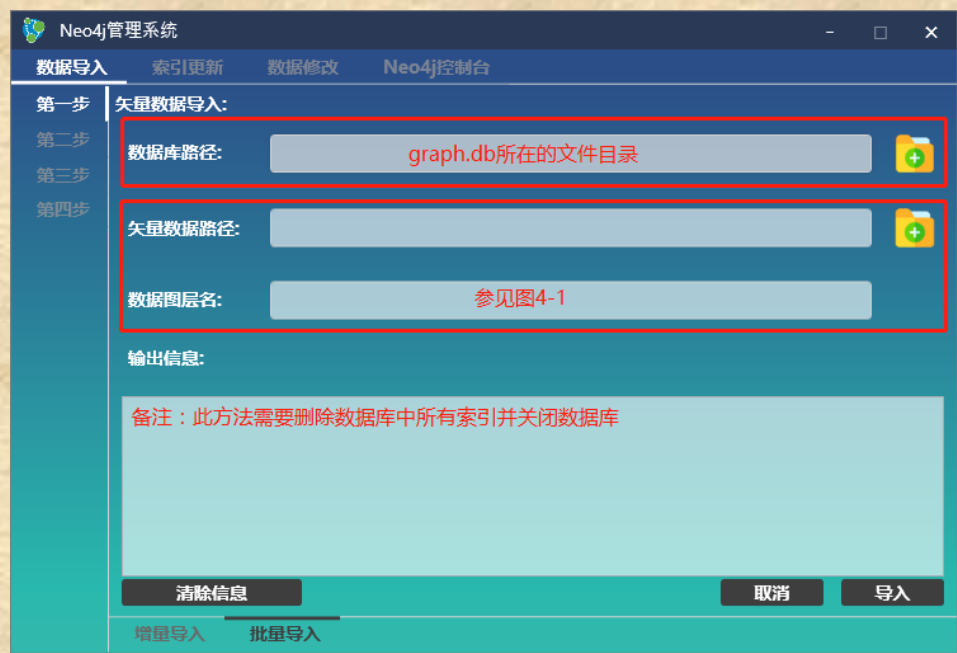


图 4-2 批量矢量数据导入

- b) 标签更新
- c) 创建关系
- d) 计算节点状态

2) 索引更新

3) 数据更新

4) Neo4j 控制台

4.2 时空数据查询

附录 1. Neo4j 数据库要素属性表

附录 2. Neo4j 数据库常用操作与语句

附录 3. Neo4j 数据库配置与环境设置

数据库配置部分

1. 开启 Neo4j 数据库远程访问

配置 Neo4j.conf 文件，修改为


```
# With default configuration Neo4j only accepts local connections.
# To accept non-local connections, uncomment this line:
dbms.connectors.default_listen_address=0.0.0.0

# You can also choose a specific network interface, and configure a non-default
# port for each connector, by setting their individual listen_address.

# The address at which this server can be reached by its clients. This may be the server's IP address or DNS name, or
# it may be the address of a reverse proxy which sits in front of the server. This setting may be overridden for
# individual connectors below.
#dbms.connectors.default_advertised_address=localhost

# You can also choose a specific advertised hostname or IP address, and
# configure an advertised port for each connector, by setting their
# individual advertised_address.

# Bolt connector
dbms.connector.bolt.enabled=true
#dbms.connector.bolt.tls_level=OPTIONAL
dbms.connector.bolt.listen_address=0.0.0.0:7687

# HTTP Connector. There must be exactly one HTTP connector.
dbms.connector.http.enabled=true
dbms.connector.http.listen_address=0.0.0.0:7474

# HTTPS Connector. There can be zero or one HTTPS connectors.
dbms.connector.https.enabled=true
dbms.connector.https.listen_address=0.0.0.0:7473
```

重新启动服务器

```
C:\Users\Administrator>neo4j restart
警告: 正在等待服务“Neo4j Graph Database - neo4j <neo4j>”停止...
Neo4j windows service stopped
警告: 正在等待服务“Neo4j Graph Database - neo4j <neo4j>”启动...
Neo4j windows service started
```

2. 配置页面缓存（推荐 20G）

```
# The amount of memory to use for mapping the store files, in bytes (or
# kilobytes with the 'k' suffix, megabytes with 'm' and gigabytes with 'g').
# If Neo4j is running on a dedicated server, then it is generally recommended
# to leave about 2-4 gigabytes for the operating system, give the JVM enough
# heap to hold all your transaction state and query context, and then leave the
# rest for the page cache.
# The default page cache memory assumes the machine is dedicated to running
# Neo4j, and is heuristically set to 50% of RAM minus the max Java heap size.
dbms.memory.pagecache.size=20g
```

3. 配置 APOC 以及 Sptial 插件

Neo4j.con 文件中添加如下代码

```
# APOC Security
dbms.security.procedures.unrestricted=apoc.*
```

4. 配置 NetTopologySuite（NuGet）

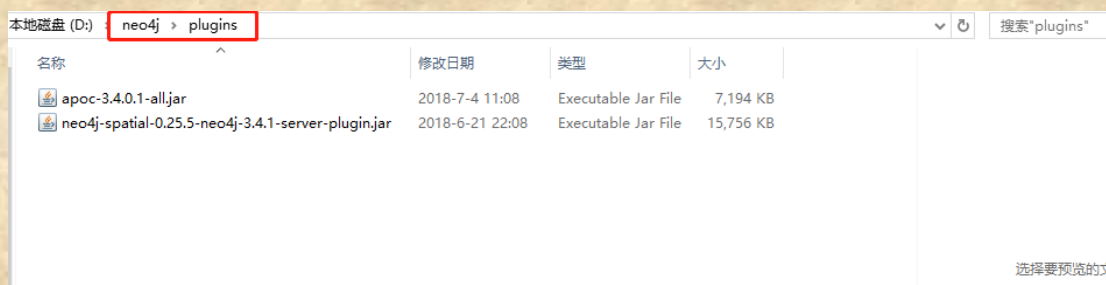
<https://www.nuget.org/packages/NetTopologySuite/1.15.0-pre063>

Install-Package NetTopologySuite -Version 1.15.0-pre063

备注：NetTopology 1.14.0 版本才能进行 WKB 转换（新版本方法有修改）

Install-Package NetTopologySuite -Version 1.14.0

5. Spatial 插件配置方法



备注：<http://neo4j-contrib.github.io/spatial/>参考文档

6. Neo4j 不支持时间的处理办法（APOC 插件）cai'liao

方法一

```
apoc.date.format(timestamp,'s','yyyy-MM-dd HH:mm:ss','CTT') (CTT 为上海时代码)
```

方法二

```
Datetime("2019-02-20T15:48:00"); //Neo4j 3.4.0 以上支持 Datetime 函数
```

7. ShpaeFile 包含中文，导入 Neo4j 乱码解决方法

需要设置 Arcgis 默认保存 Shp 的编码为 UTF-8:

打开注册表，定位到 HKEY_Current_User\Software\ESRI\，添加 Common 项，在 Common 项下面新建 CodePage 项。然后在 CodePage 项中（注册表右侧）添加一个字符串（REG_SZ），名称： dbfDefault，键值： oem（或者 UTF-8）。

第一章数据库常用操作语句

1. Shp 文件导入后，设置节点的标签

```
match (n)-[:RTREE_ROOT]-()-[:RTREE_CHILD*]->()-[:RTREE_REFERENCE*]->(m) where
n.layer=~".*RainStormEvent.*" set m:xxx
```

备注：

a) 根据要素名替换 RainStormEvent 部分

b) XXX 替换为要素名

2. 导入节点数据之间的关系

```
USING PERIODIC COMMIT 1000
```

```
LOAD CSV WITH HEADERS FROM "file:///xxx.csv" as line
```

```
match(from:xxxNode{STID:line.from_id}),(to: xxxNode {STID:line.to_id})
```

```
merge (from)-[r:SRelationship{StateAction:line.property}]->(to);
```

关系文件格式

PID (可省略)	from_id	to_id	property
-----------	---------	-------	----------

备注:

- a) xxx.csv: neo4j 目录下, import 目录中关系文件, 根据实际替换;
- b) xxxNode: 节点标签, 根据实际替换

3. 计算节点状态类型

```
MATCH (p:xxxNode)
```

```
WITH p, size ((p)-[:SRelationship]->()) AS outDegree, size ((p)-[:SRelationship]-()) AS inDegree
```

```
WHERE inDegree > 1 AND outDegree = 0
```

```
set p.Type=3;
```

五种状态定义如下:

- 1) 产生 (0): inDegree=0 AND outDegree=1;
- 2) 消亡 (1): inDegree=1 AND outDegree=0;
- 3) 分裂 (2): inDegree=1 AND outDegree>1 or inDegree=0 AND outDegree>1
- 4) 合并 (3): inDegree>1 AND outDegree=1 or inDegree>1 AND outDegree=0
- 5) 合并分裂 (4): inDegree>1 AND outDegree>1
- 6) 发展 (5): inDegree=1 AND outDegree=1

备注:

- a) xxxNode: 节点标签, 根据实际替换;
- b) SRelationship: 节点之间的关系, 根据实际替换;
- c) p.Type: 节点类型, 根据五种状态 0~5 确定